

FREE
DOS 6 Reference Card

SPECIAL ISSUE!

FROM THE EDITORS OF

DOS RESOURCE
GUIDE

50
PROGRAMS
INSIDE

JUNE 1994

SPECIAL #4

\$4.95

CANADA \$5.95

An IDG Communications Publication

MAXIMUM DOS

**HIGH-POWERED
TOOLS AND
TECHNIQUES**

HELP YOU TO:

- Fine-Tune
AUTOEXEC.BAT
and CONFIG.SYS
- Reclaim Memory

- Speed Up Your
Hard Drive
- Protect
Your Data



COMPANION
**Disk
Offer**

pg 79



1.5 Million and counting

Of course, we're referring to the 1.5 million WordPerfect users who are better at creating documents since upgrading to SIX.O.

Upgrade today, and you too can benefit from these features that are guaranteed to improve the way you work:



Dynamic new graphical interface with WYSIWYG editing



Powerful spreadsheet capabilities including AutoFill



Enhanced font support



Powerful Graphical Ruler



Drag and drop graphics handling



Automatic envelope addressing with bar code printing



Nine document viewing and editing



Fax support



Easier and faster to learn and use

"WordPerfect 6.0 is a must-buy upgrade for current users
and an excellent choice for newcomers."



November 9, 1993
WordPerfect 6.0 for DOS

"It ranks as the most powerful application ever written for DOS,
with conveniences that rival a fully equipped Windows setup."

Call **800-526-3089** to upgrade today!



Don't Miss Out!

SPECIAL ISSUE #3

From the Editors of *DOS RESOURCE GUIDE* !

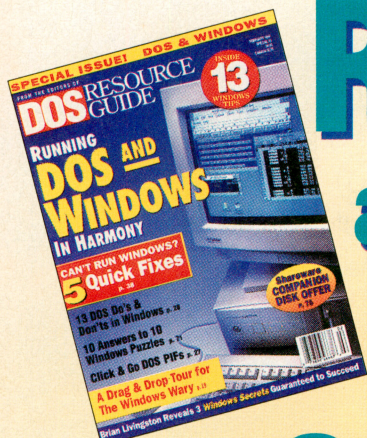
RUNNING DOS and WINDOWS

and a Great

TIME
SAVING

COMPANION DISK

Containing shareware programs that will merge
the worlds of DOS and Windows in ways you never thought possible!



DID YOU KNOW?

- Windows actually makes DOS programs run better?
- Windows renders many DOS functions and commands obsolete?
- Your DOS programs can use Windows to exchange data?
- Your DOS programs can use Windows to run tasks invisibly while you work on something else?

FIND OUT HOW TO:

- Set up your computer to use DOS and Windows together!
- Multitask your DOS programs in only minutes!
- Write custom PIFs just for your program!
- Use the hidden power of Windows' built-in software!
- Buy a computer tailored for Windows!

ORDER BOTH AND SAVE \$5.00!

AS4RW

QTY.		PRICE	CIRCLE*
	 Send me BOTH RUNNING DOS AND WINDOWS and COMPANION DISK for	\$14.90	5.25" / 3.5"
	 Send me just the SPECIAL ISSUE RUNNING DOS AND WINDOWS	\$5.95	
	 Send me just the COMPANION DISK for	\$13.95	5.25" / 3.5"
	TOTAL		

Canada, Mexico, and Foreign Surface add \$5.50 per issue/disk ordered (includes applicable tax). Airmail add \$9.50. Prepayment is required on all orders. U.S. funds on U.S. banks only. Magazines and disks will be shipped separately. Please allow 2-3 weeks for delivery.

* Important: If you are ordering a disk, please remember to specify size.

☐ Check or money order enclosed (payable to *DOS Resource Guide*)

Charge my: ☐ VISA ☐ AmEx ☐ Discover ☐ MC

Card # _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

MAIL TO: *DOS Resource Guide*, P.O. Box 672, Mt. Morris, IL 61054

DOS/Windows Companion Disk:

Here are just a few of the programs you'll receive!

WinCLI. A command line interface (like the DOS prompt) for Windows. You can launch Windows & DOS programs, or manipulate files with COPY, DELETE, RENAME, and 20 more commands. WinCLI works in all Windows modes so even a 286 can run a DOS prompt in a window!

EDOS. An amazing program that makes your DOS session "Windows-aware," whenever you run DOS in Windows' 386 enhanced mode. EDOS uses no conventional memory. Once you become accustomed to EDOS, you'll never want to be without it.

Outside! Certain DOS programs, especially high-resolution games, don't run well under Windows 386 Enhanced Mode. Outside! is a Windows program that terminates Windows, gives your DOS program the full resources of your PC, then reloads Windows when you exit the DOS program.

PLUS; More Windows/DOS utilities and a BONUS PROGRAM, **ALIEN FORCE.**

Call **1-800-349-7327**

MAXIMUM DOS

T A B L E O F C O N T E N T S

- 7 MAKING THE MOST OF CONFIG.SYS** *by Hardin Brothers*
Even the fastest processor or cutting-edge hardware won't maximize your computing if CONFIG.SYS isn't set up correctly. Follow these tips for fine-tuning your most important configuration file.
- 17 HOW TO STREAMLINE AUTOEXEC.BAT** *by Hardin Brothers*
Could your AUTOEXEC.BAT file stand some improvement? Chances are that, with a little tweaking, this master batch file can increase memory availability and make your programs run faster.
- 22 MAXIMUM MEMORY, MINIMUM FUSS** *by Dan Gookin*
The biggest problem with memory management is how to free up more RAM without losing your sanity first. The solution is to run a memory optimizer, and DOS 6's MemMaker utility is one of the best.
- 25 RUNNING IN THE FAST LANE** *by John M. Goodman*
The communication between your computer's CPU and its hard disk drive is a key factor in overall performance. You can boost disk speed with four MS-DOS commands that manage data more quickly and efficiently.
- 30 TUNING UP SMARTDRIVE** *by Stan Miastkowski*
DOS's built-in SMARTDrive disk-caching utility can make your system run smarter and faster. Here's how to customize it for your setup.
- 36 SCANDISK, AN MS-DOS SELF-STARTER** *by Doug Lowe*
MS-DOS 6.2's disk-repair program, ScanDisk, is a vigorous replacement for CHKDSK. However, it insists on your undivided attention. Boost ScanDisk's efficiency by setting it up to run unattended.
- 41 OUT OF SIGHT, OUT OF MIND?** *by John Wolfskill*
Is your hard drive in the best shape it can be? You'll never forget to run disk maintenance with these programs that remind you to launch MSBACKUP, DEFRAG, CHKDSK, or ScanDisk on a regular schedule.
- 47 USING DOSKEY COMMAND MACROS** *by Craig Menefee and Nick Anis*
DOSKEY macros are like power tools for your computer. Learn more about these time- and error-saving commands with this excerpt from the book Harnessing DOS 6.0: Batch File and Command Macro Power.
- 53 SPECIAL SECTION: MAXIMUM DOS TOOL KIT**
Take control of your PC! We've included 24 pages of our best batch programs, Debug scripts, and command techniques to make your computing time more productive.
- 88 THE MAXIMUM DISK FOR MAXIMUM DOS** *by Jeff DeTray*
There's no need to type in all the listings in this special issue of DOS Resource Guide. The Maximum DOS Companion Disk contains all the programs in this issue, as well as three bonus shareware programs.

TIPS

- | | | |
|--------------------------------------|------------------------------|--|
| 26 Don't Forget the Also-Rans | 34 Multiple SMARTs | 43 Shorten the PATH, Increase Speed |
| 32 Best Buffers | 35 Maximum SMARTDrive | |

ABOUT THIS Special Issue

by David Essex

To Windows converts, the phrase "Maximum DOS" must seem an oxymoron. But to the millions of DOS users who continue to frequent the C:\> prompt, and who look forward to the next upgrade from Microsoft, making the most of DOS remains an important goal. The fact is, DOS itself contains many of the best tools for improving overall PC performance.

To address this continuing need, every feature article in *Maximum DOS* contains a noted DOS expert's no-nonsense advice on tuning up a critical part of the DOS/hardware machine. Here are a few highlights.

We start with those often frustrating DOS configuration files, CONFIG.SYS and AUTOEXEC.BAT—also the two most important files for improving your PC setup. Subsequent features cut right to the chase with tips on how to squeeze extra speed and space out of your memory and hard disk. You also learn how to make disk and memory work in tandem with SMARTDrive, the disk-caching program that is perhaps DOS 6's most impressive speed-enhancing utility.

Hardware tweaking amounts to little if the biggest performance bottleneck is *you*. That's why we've asked former *PC Magazine* Senior Technical Editor Robert Hummel to comb previous *DOS Resource Guide* issues for our best batch files, Debug scripts, and QBasic programs to form the *Maximum DOS* Tool Kit. These are the directory daredevils and keyboard cure-alls that make life around the DOS prompt more livable. Numerous readers have asked for such a collection; the best are all here for you to type in, or you'll find them ready-to-run on the Companion Disk and on our BBS (603-924-3181).

A word on DOS-version compatibility is in order. Except where noted, all listings and tips work with both MS-DOS 5 and 6.x. Some also work with earlier DOSs, but we sacrificed uniform backward compatibility to get the most from the later, utility-rich upgrades. Older listings made obsolete by DOS 5 or 6.x were dumped; only the diehards remain. We trust you'll agree the trade-off was a fair one. ■

David Essex is *DOS Resource Guide*'s Senior Editor. Previously, he was an editor and writer at *Portable Office* and *PC Resource*.

DOS Resource Guide welcomes letters, complaints, and submissions from readers. The easiest way to reach the editors is the U.S. mail: DOS Resource Guide, 80 Elm St., Peterborough, NH 03458.

All letters to the editor and questions are understood to be submitted for publication unless they are otherwise indicated. You can reach our staff electronically over CompuServe at 75300,2361. Please remember to include your complete address and a daytime phone number on your correspondence.

Also, you can reach the editors through the DRG Bulletin Board System (603-924-3181). To connect, set your modem and software to 8 data bits, no parity, and 1 stop bit. DRG's BBS lists all of the QBasic and shareware programs mentioned in these pages. Shareware listed on the DRG BBS are

How to Contact The Editors

products protected by copyright law. You are welcome to try these programs. If you find them useful, you are asked to register and pay the applicable fees to the programs' respective owners.

DRG encourages the submission of DOS tips, QBasic programs, and batch-file articles from readers. Please submit all material on disk in ASCII format together with a double-spaced printed copy to our offices. The magazine cannot be responsible for the return of these manuscripts, unless an appropriate mailer and return postage are enclosed.

DRG makes every effort to verify its articles and programs, but the magazine assumes no responsibility for any damages because of errors or omissions in our articles. Corrections to listings or articles are printed as soon as possible.

DOS RESOURCE GUIDE

MAXIMUM DOS

EDITOR

David Essex

COPY/PRODUCTION EDITOR

Hilory Wagner

ARTICLE EDITORS

Patricia Cummings, Beth S. Jala,
Marilyn McMaster

TECHNICAL EDITORS

Hardin Brothers, John M. Goodman,
John Wolfskill

CONTRIBUTING EDITORS

Dan Gookin, Robert Hummel,
Doug Lowe, Stan Miastkowski

ELECTRONIC PUBLISHING

Lindy P. Greenwood

DESIGN/PRODUCTION CONSULTANTS

Marilyn Fletcher
Dianne Ritson

PUBLISHER

Stephen Robbins

NATIONAL ACCOUNTS MANAGER

Bob Byrne
(908) 972-8834

SALES REPRESENTATIVE

Linda M. Guyette
(603) 924-0130, (800) 441-4403

Business Computer Publishing Inc.

PRESIDENT

Jeff DeTray

EDITORIAL DIRECTOR

Michael J. Comendul

CIRCULATION MANAGER

Cathy Carabello

CIRCULATION ANALYST

Kim Smith

ADMINISTRATIVE ASSISTANT

Mary Hartwell

NEWSSTAND CIRCULATION

Kemco Publisher Services (603) 924-0224

CUSTOMER SERVICE

(800) 349-7327

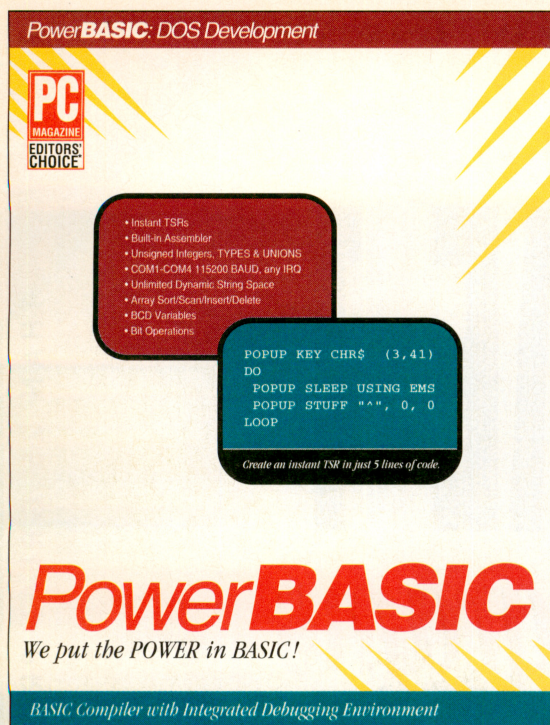
DOS Resource Guide (ISSN: 1056-7364) is published six times a year (excluding special issues) by Business Computer Publishing Inc., an IDG company. Business offices at 80 Elm St., Peterborough, NH 03458, (603) 924-0100. Entire contents copyright 1994 by Business Computer Publishing Inc. No part of this publication may be printed or otherwise reproduced without written permission from the publisher.

U.S. subscription rate is \$23.70, one year; \$44.00, two years; Canada (including GST) and Mexico, \$29.70; foreign surface, \$41.70, one year; foreign air-mail, \$53.70, one year. Payable in U.S. funds drawn on a U.S. bank. Prepayment is required on all foreign subscriptions. Second-class postage is paid at Peterborough, NH, and additional offices. POSTMASTER: Send address changes to *DOS Resource Guide*, P.O. Box 669, Mount Morris, IL 61054.



DOS Resource Guide is a publication of International Data Group, the world's leader in information services on information technology.

PRINTED IN THE U.S.A.



"... DOS (BASIC) developers would do better to look beyond Microsoft.™"

PC Magazine, September, 1993.

PowerBASIC Awarded Editor's Choice

"These are tough criteria, and almost every product we looked at came up short. But not PowerBASIC, Version 3.0. Our Editor's Choice in the DOS category, it delivered in almost every regard. Its impressive feature set includes a wide selection of data types (including user-defined), support for unlimited string space, and the ability to create and manipulate large bit arrays. Its powerful editor and debugger make for fast development, and its in-line assembler is unique to the reviewed products. The fast compiler generates highly efficient executables for a wide variety of CPU/FPU's. The package also supports TSR Development. While price may not be a major issue for a professional development tool, the \$149 price can't help but be attractive, and the product's documentation and support are exemplary." PC Magazine, September, 1993.

Upgrade to . . .

PowerBASIC 3.0

Struggling with slow interpreters like QBasic or BASICA?™ Unhappy with bugs and out-dated performance of other compilers? Then it's time to upgrade your code to PowerBASIC™ 3.0! Source code is 99% compatible with QuickBasic, so the move up to PowerBASIC couldn't be easier. Blazing speed ... precise calculation ... and a myriad of advanced features make PowerBASIC 3.0 the upgrade of choice.

Instant TSRs

Create Pop-Up TSRs with just five lines of code. A 600K TSR can "pop-down" to a memory image of just 4K, with automatic memory swapping to EMS or virtual disk. Plus, two-way communication between multiple TSRs is offered.

Built-in Assembler

Hand optimize critical sections of your code for absolute performance. Whether 1 line or 1000, mix BASIC and Assembler — the decision is yours.

Advanced Data Types

Unsigned integers (byte, word, double word). User-defined TYPES and UNIONS. Quad-word, 64-bit integers. Extended precision, 80-bit floats. Dynamic, fixed-length, and flex strings.

BCD Variables

Eliminate round-off errors, even those with currency variables. You aren't forced to use 4 decimal places. Rather, choose from 0 to 18 digits — always with automatic rounding to the precise level needed.

Serial Communications

COM1 through COM4. BAUD rates to 115200. Non-standard IRQ lines are all supported.

Advanced Features

Unlimited dynamic string space. Array Sort, Scan, Insert, Delete. Procedural floating point option for fast non-8087 performance. BIT Shift, Rotate, Test, Set, Reset, Toggle. Implied bit arrays up to 512K bits. Conditional compilation (\$IF, \$ELSE, \$ENDIF). Text viewports. EXIT FAR allows multi-level sub/function exit (beats the LONGJMP in C, any day).

- ☐ **YES!** Upgrade me for \$119 (reg. \$149).
- ☐ No, please send me a FREE demo disk first.

PowerBASIC, Inc.

1199 Forest Ave., Suite 100
Pacific Grove, CA 93950
(408) 372-8000 — FAX (408) 372-8005
(800) 780-7707 — BBS (813) 625-1721

Offer expires 6/15/94.

A HOT NEW DISK FROM THE EDITORS OF DOS RESOURCE

MORE THAN
25 QBasic
PROGRAMS
ON DISK!

SUPER UTILITIES, CLASSIC GAMES, HANDY APPLICATIONS

The DOS Resource Guide
Software Library

VOLUME 1

QBasic Programs

More than 25 QBasic Programs from
the pages of DOS Resource Guide.
Includes Super Utilities, Classic Games,
Handy Applications! Save typing time!
All programs Ready-to-Run using QBasic*
Complete instructions right on the Disk!

Here are just a few of the programs in Volume 1 of the **DRG Software Library**:

UTILITIES **Colors** - Helps you select the perfect color scheme for your batch files and QBasic programs. **Scramble** and **Descramble** - Protect files from prying eyes with this pair of programs. **Clean** - Delete unwanted files from your hard drive quickly and safely. **Printer Primer** - The easy way to set up your printer for special effects.

GAMES **CRAM** - Maneuver an ever-growing worm inside an ever-shrinking box. **Diamond Pong** - A QBasic version of the classic Pong video game. **Letter Blaster** - Helps you improve your typing skills while you blast letters to smithereens!

APPLICATIONS **Amortizer** - Calculate loan payments and interest before you borrow. **Calendar** - Print a calendar for any year. **Names** - Your own electronic name and address file.

BONUS UTILITY **Portal** is an easy-to-file transfer program that lets you copy files between two computers via their serial ports. It's simple and reliable, and it's included **FREE** on your DRG Software Library disk!

ORDER NOW! JUST \$9.95! ORDER NOW! CALL 1-800-349-7327

MORE THAN 25 GREAT PROGRAMS NEVER BEFORE ON DISK!

*QBasic programs require QBasic, the programming language included with MS-DOS 5.0 and later. The Bonus program *Portal* is a stand-alone program that does not require QBasic.

yes, send me **VOLUME 1**
of the DRG Software Library

Quantity Price Ea.

	\$9.95
Shipping and Handling per Disk	\$1.95
Total per Disk	\$11.90

Important!
Please Circle
Disk Size

3.5" 5.25"

Total
Enclosed:

☐ Check or money order enclosed (payable to DOS Resource Guide)
Charge my: ☐ VISA ☐ AmEx ☐ Discover ☐ MasterCard

Card # Exp. Date

Signature

Name (print)

Address

City

State

Zip

Prepayment is required on all orders. U.S. funds on U.S. banks only.
Airmail add \$5.50. Please allow 2-3 weeks for delivery.

1-800-349-7327 Or mail to: DOS Resource Guide
PO Box 672, Mt Morris, IL 61054

MAKING THE MOST OF CONFIG.SYS

To create the best possible atmosphere for your computing sessions, you may need to refine—and redefine—the ground rules.

by Hardin Brothers

DOES YOUR COMPUTER DO YOUR BIDDING AS QUICKLY AS POSSIBLE, with a smile and a nod? Or does it lumber its way through your work, with a chip on its shoulder? If, despite an adequately fast processor and up-to-date peripherals, your system seems to plod along, don't blame the hardware gremlins. □ To a large extent, *you* control your PC's personality—by creating and maintaining the configuration files that tell the system how to organize itself each time you

boot up. If you use these files correctly, you'll get as much power as possible from your hardware. If you aren't careful, however, your configuration files can drag down your computer, slowing disk operations or even halting program execution.

This article explains how to fine-tune the most important configuration file, CONFIG.SYS. A follow-up piece, "How to Streamline AUTOEXEC.BAT" (see page 17), explains how to do the same for AUTOEXEC.BAT, the second-most important file on your hard disk.

What CONFIG.SYS Does

Your CONFIG.SYS file has three purposes. First, it lets you extend your machine's capabilities by installing "devices" that DOS originally wasn't built to handle. Second, you use it to set several DOS options and,

third, to control the way in which DOS uses memory and other resources. If you remember that CONFIG.SYS has these three distinct purposes, you can understand CONFIG.SYS more fully and modify it with greater confidence and success.

When you boot your computer, it performs internal diagnostics and loads part of DOS, which then looks for a file called CONFIG.SYS in the root directory of the boot drive. If the file is found, it is used to set up the computer and load a full-fledged version of DOS. If the file isn't found (perhaps because you accidentally deleted it), or if you tell DOS to ignore the file (see the sidebar "New Start-up Options in MS-DOS 6.x"), your computer boots up in DOS's default configuration, without installing any "extra" devices.

The default configuration values are Microsoft's

Take the Dread Out of DOS!

SUBSCRIBE TO DOS RESOURCE GUIDE:

Get 6 issues of
**DOS Resource Guide for
\$23.70.*** (That's just \$3.95 an issue!)

Save even more on 12 issues:
\$44.00.* (That's 25% off the cover price!)



1-800-349-7327

DOS RESOURCE GUIDE, the bimonthly periodical packed with tips, tutorials, and information on all aspects of PC computing, is your antidote for computing confusion.

Beginners will find step-by-step help for copying files, creating and removing directories, and practical advice on using DOS's wildcards to cut down on keystrokes.

More advanced users will learn how to use batch files, and how to automate tedious activities such as backing up large groups of files. And you'll learn to use DOS's QBasic programming language. Each issue of *DOS RESOURCE GUIDE* is filled with articles about DOS fundamentals, tips on using DOS commands, undocumented shortcuts, and batch files you can type and run.

Start today! Tap all of your computing potential by learning DOS, personal computing's most important operating system. Use the coupon, or call (toll-free):

1-800-349-7327

SAVE
UP TO
25%

SUBSCRIBE

- ☐ **YES**, Send me 6 issues of *DOS Resource Guide* for \$23.70. *Save me a buck an issue!
- ☐ **YES**, Save me even more: I'll take 12 issues of *DOS Resource Guide* for \$44.00.* That saves me 25% off the cover price!

Name: _____

Address: _____

City: _____

State/Zip: _____

☐ Payment enclosed ☐ Bill me

Charge my: ☐ MasterCard ☐ VISA ☐ AMEX ☐ Discover

Card #: _____

Expire Date: _____

Signature: _____

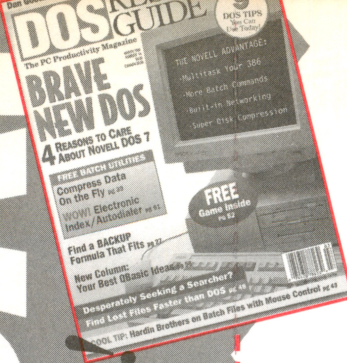
*In Canada and Mexico pay \$29.70 (12 issues \$55.00); all other foreign surface mail pay \$41.70 (12 issues \$78.00); foreign air mail pay \$53.70 (12 issues \$99.00). Foreign/Canadian orders must be prepaid in U.S. funds drawn on a U.S. bank. Please allow 6-8 weeks for delivery.

AS4DRG

MAIL TO: DOS Resource Guide, P.O. Box 669, Mount Morris, IL 61054

SUBSCRIBE TODAY!

Save up to 25% Off the Cover Price!



BUCK THE TREND! STAY WITH DOS!

☐ **YES!**

Send me 6 Issues of DOS Resource Guide for \$23.70.*
That's a **DOLLAR OFF** each issue on the newsstand!

☐ **YES!**

SAVE EVEN MORE ON 12 ISSUES! Send me 12 for \$44.00*

NAME _____

ADDRESS _____

CITY _____

STATE/ZIP _____

☐ Payment enclosed ☐ Bill me

Charge my: ☐ MasterCard ☐ VISA ☐ American Express ☐ Discover

Card# _____ Expire Date _____

Signature: _____

Please make checks payable to DOS Resource Guide.

*All Foreign/Canadian orders must be prepaid in U.S. funds drawn on a U.S. bank.

*Canada and Mexico: \$29.70 (includes GST)/2 years \$55.00.

*All other foreign (Surface Mail):

\$41.70/2 years \$78.00

*Foreign Air Mail: \$53.70/2 years \$99.00

BSP41

BUCK THE TREND! STAY WITH DOS!

☐ **YES!**

Send me 6 Issues of DOS Resource Guide for \$23.70.*
That's a **DOLLAR OFF** each issue on the newsstand!

☐ **YES!**

SAVE EVEN MORE ON 12 ISSUES! Send me 12 for \$44.00*

NAME _____

ADDRESS _____

CITY _____

STATE/ZIP _____

☐ Payment enclosed ☐ Bill me

Charge my: ☐ MasterCard ☐ VISA ☐ American Express ☐ Discover

Card# _____ Expire Date _____

Signature: _____

Please make checks payable to DOS Resource Guide.

*All Foreign/Canadian orders must be prepaid in U.S. funds drawn on a U.S. bank.

*Canada and Mexico: \$29.70 (includes GST)/2 years \$55.00.

*All other foreign (Surface Mail):

\$41.70/2 years \$78.00

*Foreign Air Mail: \$53.70/2 years \$99.00

BSP42

Give DOS Resource Guide!

Give DOS Resource Guide and save 25% off the regular rate.
Your friend gets a full year of DOS Resource Guide (6 bimonthly issues). You pay only 23.70.*

GIFT TO: RECIPIENT'S NAME _____

ADDRESS _____

CITY _____

STATE / ZIP _____

GIFT FROM: YOUR NAME _____

ADDRESS _____

CITY _____

STATE / ZIP _____

☐ Payment enclosed ☐ Bill me

Charge my: ☐ MasterCard ☐ VISA ☐ American Express ☐ Discover

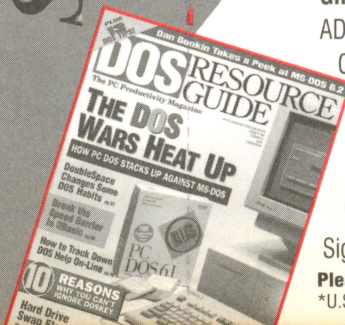
Card# _____ Expire Date _____

Signature: _____

Please make checks payable to DOS Resource Guide.

*U.S. only. Sorry, no billing available on subscriptions outside U.S.

XSP43



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 39 MOUNT MORRIS, IL

POSTAGE WILL BE PAID BY ADDRESSEE

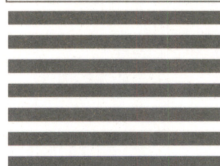
DOS Resource Guide

PO Box 669

Mount Morris, IL 61054-8100



NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 39 MOUNT MORRIS, IL

POSTAGE WILL BE PAID BY ADDRESSEE

DOS Resource Guide

PO Box 669

Mount Morris, IL 61054-8100



NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 39 MOUNT MORRIS, IL

POSTAGE WILL BE PAID BY ADDRESSEE

DOS Resource Guide

PO Box 669

Mount Morris, IL 61054-8100



NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES



Send me

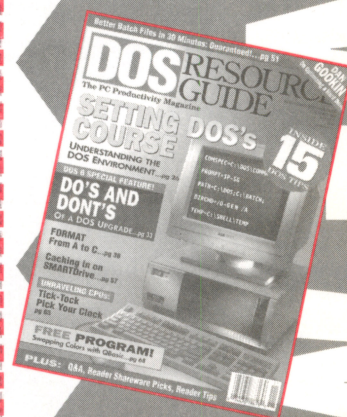
12
ISSUES

(two years)

of

DOS Resource Guide

\$44.⁰⁰



Send me

6
ISSUES

(one year)

of

DOS Resource Guide

\$23.⁷⁰

best guesses at acceptable values for all computers. You may think that these default settings should be good enough, especially if you don't use your computer for anything unusual. But they rarely are. Often, you may improve on the default settings, as well as on the settings chosen by DOS's setup program, because DOS's definition of what is standard and your definition are likely to differ.

Before you set about making any improvements, however, you must take one important precaution: creating an emergency bootup disk. Whenever you or an application's installation program alter a configuration file, you run the risk of being unable to boot from your hard disk. If problems arise, a bootable floppy disk will help you get your system running again.

To create an emergency bootup disk, format a floppy disk in drive A by typing `FORMAT /S` to tell DOS you want to create a "system" disk. Then, check to make sure that the floppy disk contains a copy of `COMMAND.COM`, and test the disk by using it to boot your computer. When you're satisfied that the disk boots properly, label it and put it in a safe place. Someday, you may be thankful you have it.

Building a CONFIG.SYS File

DOS 6.x offers one undocumented and 16 `CONFIG.SYS`-only commands; four commands that you may use in `CONFIG.SYS`, `AUTOEXEC.BAT`, or at the DOS prompt; and six commands for building a menu of start-up options. Earlier versions of DOS support fewer commands. (The sidebar "Configuration Commands Explained" lists the full complement of commands available in DOS 6.x and briefly describes their functions.)

Perhaps the two most important `CONFIG.SYS` commands are `DEVICE` and its cousin, `DEVICEHIGH`. These commands load device drivers, memory-resident programs that expand your computer's capabilities. The difference between the two commands is that `DEVICE` loads drivers into conventional memory, while `DEVICEHIGH` loads them into upper memory. Without these commands, PCs can't work with many types of software and hardware that we have come to think of as standard: memory managers, mice, RAM disks (portions of memory that function as disk drives), networks, extended and expanded memory, CD-ROM drives, external disk drives and backup devices, scanners, and a host of other add-ons.

If you have a 286, 386, 486, or Pentium computer and more than 640K of memory, the first command in your `CONFIG.SYS` file should be a `DEVICE` statement for loading a memory manager. Unless you are using a third-party memory manager such as Quarterdeck Office Systems' `QEMM` or Qualitas' `386MAX`, the first line of your `CONFIG.SYS` file should look like this:

```
DEVICE=C:\DOS\HIMEM.SYS
```

If you store `HIMEM.SYS` in a different directory, you must specify the correct path.

To fine-tune `HIMEM` for your computer, you also may need to add an option or two to the end of the line; the options are explained in detail in MS-DOS's on-line help. This line (or the equivalent line for a third-party memory manager) is so important, that it almost always appears at the beginning of `CONFIG.SYS` (for an example, see the sidebar "Sample `CONFIG.SYS` files.")

If you use a 386 or later computer with at least 1MB of RAM, you probably want to free as much DOS, or conventional, memory as possible. If you want to use DOS's own memory management tools to do this, the next line in your `CONFIG.SYS` file should read:

```
DEVICE=EMM386.EXE
```

(In DOS 5, the program is called `EMM386.SYS`.)

`HIMEM.SYS` gives DOS and programs access to extended (XMS) memory and lets DOS load itself into the High Memory Area (HMA). `EMM386` adds further memory-management capabilities, letting you use XMS memory as expanded (EMS) memory and permitting you to load memory-resident utilities (also called TSRs) and other device drivers into your system's Upper Memory Blocks (UMBs) so that you have more room in conventional memory for applications such as a word processor or spreadsheet. (For an explanation of memory terms, see the article "Maximum Memory, Minimum Fuss" on page 22.)

With memory management under control, your next job is configuring DOS. Unless you have a program that makes better use of the HMA than DOS does, the third line in `CONFIG.SYS` should move DOS into the HMA and give it control of the UMBs:

```
DOS=HIGH,UMB
```

(The two commands `DOS=HIGH` and `DOS=UMB`, when used together, accomplish the same end as `DOS=HIGH,UMB`.) If you are using a third-party memory manager, substitute the command that it recommends.

Of BUFFERS, FILES, and Other DOS Settings

Next, set the number of disk buffers that DOS may use to hold certain disk data in memory. Each buffer takes 512 bytes (.5K) of RAM for the disk data plus 16 bytes for housekeeping. Reserving buffer space improves the speed of disk read operations.

Selecting the correct number of buffers is tricky, because the proper setting depends on whether you use a disk cache, such as DOS's `SMARTDRV.EXE`; whether you use `FASTOPEN` (a dangerous utility designed to speed hard disk access); the size and speed of your hard disk; and how often you use floppy disks.

The `BUFFERS` setting may include two values, the

NEW START-UP OPTIONS IN MS-DOS 6.X

MS-DOS 6.x includes a number of new features that can help you avoid frustration while fine-tuning your CONFIG.SYS file:

Bypassing CONFIG.SYS. When the "Now starting MS-DOS..." message appears on the screen, you may press F5 to avoid executing your CONFIG.SYS file. If your CONFIG.SYS contains an error that locks up the computer during bootup, this feature lets you start your computer and use an editor to correct the error. Although this capability doesn't completely eliminate the need for an emergency boot disk, it lets you correct many boot-up problems.

Executing commands one by one. Pressing F8 instead of F5 lets you single-step through your CONFIG.SYS file (and through your AUTOEXEC.BAT if you are using DOS 6.2). You choose whether to execute or ignore each line or whether to execute or ignore all remaining commands. If you don't know which line is causing an error or lockup, one boot with F8 will tell you. Simply accept each line in turn until the error occurs.

Selectively executing commands. To have the flexibility of deciding at bootup whether to execute a CONFIG.SYS line, place a question mark (?) immediately in front of the command's equal sign. If, for instance, you want DOS to ask for confirmation before executing DOS=HIGH, edit the line to read: DOS?=HIGH. This lets you decide whether to place DOS in the HMA (assuming you have installed HIMEM.SYS or an equivalent memory manager) by accepting the line or to leave DOS in conventional memory by declining to execute it. (To check the effect of the two variations on memory use, type MEM /D/P.)

Disabling one or more lines. If you want to include a line in CONFIG.SYS but tell DOS not to execute it, you may follow the time-honored practice of preceding it with a REM command. DOS 6.x offers an alternative, however: prefacing a line with a semicolon (;). (DOS 3.3 and earlier issue a harmless

error message when they encounter REM, but this method works, because DOS makes no attempt to execute the words following REM.)

REMs and semicolons come in handy when you need to add descriptive comments to your CONFIG.SYS file, perhaps explaining why you have included other lines or changed the values in some commands:

```
REM Old setting:
REM BUFFERS=10
REM Change suggested by XYZ-Cal:
BUFFERS=20
```

But often a more important reason for using REM or a semicolon is to disable a line temporarily while you are testing variations on your CONFIG.SYS file. It is much easier to type REM plus a space at the beginning of a line than it is to erase the line and later retype it if you decide you want to put it back in CONFIG.SYS.

Creating multiple configurations. DOS 6.x's final—and most powerful—new setup feature lets you create colorful CONFIG.SYS menus and establish default selections for unattended bootups. If no choice is made within a given amount of time, DOS uses the default settings.

DOS 6.x's multiple-configuration commands are:

[COMMON], which defines a block of CONFIG.SYS commands to be executed by all menu choices;

INCLUDE, which lets you include one menu block in another and thus avoid repeating blocks of CONFIG.SYS commands;

MENUCOLOR, which defines the text and background color of a menu;

MENUDEFAULT, which selects a menu item as the default choice and sets a time-out value;

MENUITEM, which defines a name for a menu choice and the

number of file buffers and the number of "read-ahead" buffers. This line, for example, creates 40 file buffers and eight read-ahead buffers:

```
BUFFERS=40,8
```

The read-ahead setting is optional; if you omit it, DOS uses its default setting of zero. When a program, or DOS itself, reads a sector of a disk into a buffer, it fills its read-ahead buffers with the next sectors on the disk.

If you use SMARTDRV.EXE, or a similar disk cache, you don't need many buffers at all; a setting of three to five buffers and no read-ahead buffers should suffice. If you don't use a disk cache, you should base the number of buffers on the size of your hard drive. A

good rule of thumb is to use 10 buffers plus an additional 10 per 40MB of hard drive space available. That is, use 20 buffers for a 40MB drive, 30 for an 80MB drive, and so on. If the number of buffers is too high, DOS spends more time searching the buffers than reading data from the hard disk. You'll rarely need more than 50 buffers, no matter how high the capacity of your hard disk.

Read-ahead buffers are most useful if you spend a lot of time reading data from floppy disks, but only if you don't use a disk cache, which does much the same thing. Creating eight read-ahead buffers seems to work well on systems that lack a disk cache and rely heavily on floppy disks.

If the BUFFERS setting is low enough, MS-DOS loads the buffers into the HMA, but it won't split the buffers

text that will appear on the menu;

SUBMENU, which lets you create a submenu for a particular menu choice.

When you select an option from a CONFIG.SYS menu, your choice is sent to AUTOEXEC.BAT in an environment variable called CONFIG. If, for example, your CONFIG.SYS file has two menu choices, DOS and WINDOWS, you may start your AUTOEXEC.BAT file with a GOTO %CONFIG% command and have it branch to one of two labels, :DOS or :WINDOWS.

The structure of a multiple-setup CONFIG.SYS is as follows:

```
REM Demonstration CONFIG.SYS
[MENU]
NUMLOCK=OFF
MENUMCOLOR=1,15
MENUDEFAULT=DOS, 10
MENUITEM=DOS, DOS CONFIGURATION
MENUITEM=WIN, WINDOWS CONFIGURATION

[DOS]
; Place your DOS configuration commands here.

[WIN]
; Place your Windows configuration commands here.

[COMMON]
; Place the commands common to Windows and DOS here.

The corresponding AUTOEXEC.BAT would look like this:

REM Place commands for both DOS and Windows here.
GOTO %CONFIG%

:DOS
REM Place DOS setup commands here.
GOTO DONE

:WIN
REM Place Windows setup commands here.
```

:DONE

REM Place final commands for DOS and Windows here.

The CONFIG.SYS file shown above starts by turning off the Num Lock key and setting up the screen to display blue characters on a white background. It also defines the DOS configuration as the default, the one to be used if no choice is made within 10 seconds. On screen, the menu choices appear as *DOS Configuration* and *Windows Configuration* but the internal names for the choices are DOS and WIN.

There are four sections of commands in this CONFIG.SYS sample. Commands above the [MENU] line will be executed before the menu appears. Depending on the user's choice at the menu, either the commands that follow the [DOS] label or those that follow [WIN] will be executed. The commands following the [COMMON] label are executed no matter which menu choice the user makes.

The organization of the AUTOEXEC.BAT file is similar. You may place common commands at the beginning or the end. DOS uses the selection made in CONFIG.SYS to determine whether to execute the DOS or WIN section.

The only disadvantage of multiple configuration menus is that they require you to take special care when using memory optimizers such as MemMaker. One way to optimize multiple setups is to save your CONFIG.SYS and AUTOEXEC.BAT files under different names, extract the pertinent commands for the first menu choice in CONFIG.SYS and AUTOEXEC.BAT, and then run MemMaker. Merge the resulting changes and switches into your original configuration files, then extract the blocks for each of the other menu choices in turn, and repeat the process.

When you're done, rename the saved files as CONFIG.SYS and AUTOEXEC.BAT, and reboot.

—H.B.

between the HMA and conventional memory. (DR DOS, however, has this capability.) The amount of space available in the HMA varies, but usually you can get DOS to load a total of 45 buffers there (running DoubleSpace in MS-DOS 6.2 is an exception that reduces this buffers limit to 14). You may type MEM /D/P to see where the buffers are stored on your system.

Most users place a FILES command immediately before or after their BUFFERS command. This line specifies how many files may be open at one time. If you run only one DOS program at a time, set FILES to the highest number recommended by the applications you use. Database managers and spreadsheets require you to open many files at once. If you run multiple applications at the same time, using multitasking or task switching software such as Microsoft Windows or

Quarterdeck's DESQView, you may need to increase the FILES setting. You also must increase it if an application reports that it doesn't have enough file handles.

The FCBS setting is similar to FILES. It specifies the number of File Control Blocks (FCBs) that may be open. Today's programs rarely employ FCBs; unless you use very old applications, you may omit this command and let DOS use its default setting of four. Or you may save a small amount of conventional memory by employing FCBS=0.

Often, you may save additional memory using a STACKS command such as STACKS=0,0. STACKS creates a pool of temporary memory for hardware drivers to use. Most of the time, DOS and DOS programs have plenty of stack space, so you needn't make any more space available. If, however, you receive an "Internal

CONFIGURATION COMMANDS EXPLAINED

The CONFIG.SYS commands and their parameters and default values differ slightly from one version of DOS to another. If you have MS-DOS 6.x, you may use the following commands in CONFIG.SYS. If you have an earlier DOS version, see your manual or your on-line help files for information on what commands and options are available to you.

CONFIG.SYS Commands

You may use the following commands only in CONFIG.SYS:

BUFFERS=*m*[,*n*]—Establishes the number of primary buffers (*m*) and the number of secondary, or read-ahead (*n*), buffers available to DOS. When CONFIG.SYS includes a DOS=HIGH command, the buffers are stored in the HMA if space permits. Set the number of buffers low (and don't use secondary buffers) if you are using a disk cache such as DOS's SmartDrive. The default value of *m* depends on how much memory you have available and the sizes of the drives in your PC; *n* defaults to zero.

COUNTRY=*filename*—Configures DOS to use a character set other than the default; in the U.S., the default character set is the one for U.S. English.

DEVICE=*filename*—Loads a device driver into conventional memory and initializes it.

DEVICEHIGH=[/L [/S]] *filename*—Loads a device driver into upper memory (if space permits) and initializes it. If insufficient UMB space is available, this command acts the same as DEVICE=. The switches /L and /S help optimize memory; they are most effective when set by MemMaker.

DOS=[HIGH|LOW] [UMB|NOUMB]—Tells DOS whether to load

part of itself into the HMA (via the HIGH and LOW directives) and whether to maintain a link between conventional memory and the upper memory area (via UMB and NOUMB). By default, DOS is loaded into conventional memory and has no control over upper memory. (The equivalent command is DOS=LOW,NOUMB.) You may use DEVICEHIGH commands in CONFIG.SYS and LOADHIGH commands in AUTOEXEC.BAT only if CONFIG.SYS gives DOS control over UMBs. All four combinations of HIGH/LOW and UMB/NOUMB are valid.

DRIVPARM=—Sets the characteristics of a disk drive or similar device.

FCBS=*n*—Sets the number of File Control Blocks (FCBs) that DOS may have open at one time; *n* may range from 1 to 255. The default is 4, which may be higher than you need, unless you run old applications.

FILES=*n*—Sets the number of files that may be open at one time. Normally, you should permit DOS to open at least 20 files; increase this number if your applications suggest a higher setting.

INSTALL=*filename*—Loads a memory-resident (TSR) program during CONFIG.SYS processing. INSTALL commands are processed after all DEVICE commands are executed and before COMMAND.COM is loaded. Not all TSRs may be loaded with INSTALL. MemMaker can't optimize the memory used by TSRs loaded with the INSTALL command.

INSTALLHIGH=*filename*—An undocumented command that loads a TSR into an upper memory block. Because MemMaker can't optimize the memory used by the TSR, generally it's better to use AUTOEXEC.BAT's LOADHIGH command to load the program.

stack failure, system halted" message, try removing the STACKS command and accepting the default: 9 internal stacks of 128 bytes each. (If you own an original IBM PC, a PC/XT, or a PC Portable, should note that the STACKS default is 0,0.) If you need a visual reminder of this default setting, you may use the command STACKS=9,128. A few unusual hardware drivers may suggest that you allocate additional stacks and increase their size.

DOS 6.x's NUMLOCK command has a different function entirely. It lets you turn off the Num Lock key at bootup, permitting you to use the numeric keypad for cursor movement. (If you have an earlier DOS version, you can find dozens of similar shareware programs that do the same thing from AUTOEXEC.BAT or the DOS command line.)

The last important DOS configuration command is

SHELL. This command names the command processor and sets the size of the master environment. A typical SHELL command might look like this:

```
SHELL=C:\DOS\COMMAND.COM C:\DOS/E:512 /P
```

This line tells DOS to load the command interpreter, COMMAND.COM (C:\DOS indicates the location of the file), and sets the size of the master environment to 512 bytes. The master environment holds the information that is displayed if you invoke SET at the DOS prompt. Many programs look for configuration information in the environment, and some batch files add information to the environment for their own use.

/P tells COMMAND.COM to load itself as the primary command interpreter. If you want to store COMMAND.COM in a directory other than the root, if you use an

LASTDRIVE=*n*—Sets the highest drive letter available. By default, the last logical drive in DOS 6.x is the one after the last drive that DOS found or created while your PC booted. If you make heavy use of SUBST or INTERLINK, or if your PC is connected to a peer-to-peer network, you may need to use this command.

NUMLOCK=[ON|OFF]—Turns the Num Lock key on or off. The default is to leave Num Lock in the state set by your BIOS (usually it turns on Num Lock). This command is handy if you use multiple-configuration menus and want to move the cursor bar with the numeric keypad.

SHELL=*filename* [*parameters*]—Sets the name and path to the command interpreter (usually COMMAND.COM). You may use the parameters to create additional environment space, name a start-up file other than AUTOEXEC.BAT, and so on.

STACKS=*n,s*—Sets the number and size of stacks that DOS maintains for hardware interrupts. *n* is the number of stacks; *s* is the size of each stack. Valid values for *n* are zero or 8 through 64; valid values for *s* are zero or 32 through 512. On most computers, the default settings are 9 and 128, respectively, but often you may save memory by using a STACKS=0,0 command.

SWITCHES=—Sets special DOS parameters. You seldom need this command unless, for security reasons, you want to disable the CONFIG.SYS bypass options.

command*?=*setting—Forces DOS to prompt the user before executing a CONFIG.SYS command.

;—Identifies a line as a remark and prevents DOS from attempting to execute it. Place the semicolon at the beginning of the line.

General-Purpose Configuration Commands

You may use the following commands in CONFIG.SYS, AUTOEXEC.BAT, and at the command prompt:

BREAK=[ON|OFF]—When this command is turned on, DOS checks for Ctrl-C and Ctrl-Break more often. By default, BREAK is off. If you use this command in AUTOEXEC.BAT or at the command prompt, omit the equal sign.

REM—Identifies a line as a remark, which should not be executed. Place REM, followed by a space, at the beginning of a line.

SET *variable*=*string*—Creates an environment variable. Prior to DOS 6.x, this command couldn't be used in CONFIG.SYS. You may wish to use SET= when you include multiple configurations in your CONFIG.SYS file. You then can test for the value of the variables in your AUTOEXEC.BAT file to select commands that should be executed for each possible configuration.

VERIFY=[ON|OFF]—When turned on, this command tells DOS to read back each disk sector after it is written. DOS checks whether the data sector is readable; it doesn't check whether the data on disk matches the data that was written. The default setting is off. If you use this command in AUTOEXEC.BAT or at the command prompt, omit the equal sign.

Alternate Configuration Commands

For a list of the commands you may use to offer several menu options in CONFIG.SYS, see the sidebar "New Start-Up Options in MS-DOS 6.x."

—H.B.

alternate command interpreter (such as JP Software's 4DOS—available on this issue's Companion Disk—or NDOS from Symantec's The Norton Utilities), or if you need to increase the size of the master environment, you must use SHELL.

The Remains of the Day

Most of the other CONFIG.SYS commands have limited value and are seldom used. BREAK, for example, slows down the computer slightly by checking for Ctrl-Break and Ctrl-C more often than usual. VERIFY forces DOS to check each disk sector as it is written, and therefore slows down disk writes. (When you use VERIFY, DOS checks whether it can read the sector; it doesn't check whether the correct data was written to the disk.) You may set VERIFY from the DOS prompt, but verification is most often used when you

specify the /V switch with COPY and XCOPY.

DOS 6.x's SWITCHES command also is of limited value. You may use it to speed up your boot sequence slightly; however, in so doing, you give up some of DOS 6.x's new safety features. You also may use it if you run Windows 3.0 and want to move the support file WINA20.386 out of your root directory.

LASTDRIVE tells DOS which drive letters to make available to physical devices, SUBST, and DOS 6.x's INTERLINK file-transfer utility. In DOS versions 2.x through 5, you must use this command if you want to use drive letters higher than E. DOS 6.x, however, always gives you one drive letter above your highest physical device. If that's sufficient, you may omit this command.

COUNTRY, which uses the support file COUNTRY.SYS, sets the format that DOS uses for its time, date, and

SAMPLE CONFIG.SYS FILES

Your computer, computing style, and set of applications are unique. You need to modify your CONFIG.SYS to include the best possible combination of commands and options for your work. You may modify the CONFIG.SYS that the DOS Setup program creates, or you can work from one of my sample programs, adding the commands you need, deleting the lines you don't need, and running MemMaker or an equivalent program until your computer has as much conventional memory as you need and seems to run as efficiently as possible.

The examples below assume you have a 386 or higher processor and at least 1MB of memory. If you have 640K of RAM or less, your choices are more restricted, because you can't use high memory, upper memory blocks, or many of the hints given in the main article. If you don't know how much memory you have, run MEM to find out.

My first sample CONFIG.SYS is for a simple system that runs only one DOS application at a time. You might start with a CONFIG.SYS such as this one if you normally run, one at a time, a simple word processor or spreadsheet, or if you sometimes run a complex DOS-based game:

```
;Menu commands (for DOS 6.X) go here.
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH
FILES=15
BUFFERS=10
```

This second, more complex, CONFIG.SYS uses UMBs to store several DOS utilities, making better use of memory:

```
;Menu commands (for DOS 6.X) go here.
REM Memory management first:
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=EMM386.EXE NOEMS
DOS=HIGH,UMB

REM Configure DOS:
FILES=30
BUFFERS=30
FCBS=0
STACKS=0,0
SHELL=C:\COMMAND.COM /E:512 /P

REM Now add device drivers:
DEVICEHIGH=C:\DOS\ANSI.SYS
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 256 /E
DEVICEHIGH=C:\DOS\SETVER.EXE
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
```

This CONFIG.SYS assumes that you'll load SMARTDRV or a similar disk cache in your AUTOEXEC.BAT file, that you want to set aside 512K of memory for a RAM disk, and that you're using DoubleSpace to increase the capacity of your hard disk. Notice that the last line doesn't enable DoubleSpace but merely moves its resident code out of conventional memory and into a UMB. Depending on your system, you also may be able to use AUTOEXEC.BAT's LOADHIGH command to move several TSRs into upper memory.

If you use a CONFIG.SYS file such as this one, make sure you run MemMaker after you test CONFIG.SYS and AUTOEXEC.BAT. Where possible, it will add the necessary /L and /S switches to each DEVICEHIGH and LOADHIGH command, ensuring that you free as much conventional memory as possible.

—H.B.

currency displays. You need it only if you don't want your computer to use the standard U.S. English formats.

The final command in this category is DRIVPARM, one of the most misunderstood configuration commands. It modifies the way DOS sees a floppy drive, overcomes some limitations of old drives, and creates "phantom" drives. Either the DRIVPARM or DOS device driver DRIVER.SYS can handle these tasks (see the sidebar "Flipping Floppies with DRIVPARM and DRIVER.SYS").

Further Conservation Measures

The remaining two configuration commands—INSTALL and DOS 6.x's INSTALLHIGH—let you load memory-resident programs from your CONFIG.SYS file. If, instead of using LOADHIGH to install a TSR into conventional memory from AUTOEXEC.BAT, you want to install it from CONFIG.SYS, you may use INSTALL. INSTALLHIGH loads ordinary TSRs into upper memory. The advantage of using INSTALL and INSTALLHIGH is that they save a little memory, because the TSRs they load don't get a copy of the environment, except when your CONFIG.SYS has menus or a SET command. Unfortunately, however, INSTALL can't load all TSRs successfully, and INSTALLHIGH doesn't recognize the /L /S parameters, as DEVICEHIGH does. As a result, if you use INSTALLHIGH, you can't use DOS's automatic memory optimizer, MemMaker, to find the best order in which to load TSRs into upper memory.

If you have a 386 or higher processor, at least 1MB of memory, and a memory manager such as EMM386.EXE, generally it's better to use AUTOEXEC.BAT's LOADHIGH command to load a TSR into upper memory than to use INSTALL or INSTALLHIGH in CONFIG.SYS. On the other hand, if you have an 8088 or 80286 computer, you'll save some memory by using INSTALL to load as many TSRs as possible. But the savings will be modest—probably 100 or fewer bytes per TSR (the exact savings depends on the size of your environment).

Still, you may achieve far greater savings—either in memory or in hard disk space—by taking a careful look at device drivers you load at bootup and scrutinizing the contents of your \DOS directory. For example, the DOS shell, WordPerfect, Microsoft Windows, and several other applications contain their own mouse drivers. If you don't need to use your mouse in any other applications, you may eliminate the line `DEVICE=MOUSE.SYS` from `CONFIG.SYS` (or delete the `MOUSE` command from your `AUTOEXEC.BAT`). And most users may safely wipe `EGA.SYS` from their hard disks and `DEVICE=EGA.SYS` from their `CONFIG.SYS`, because the program's only function is helping the DOS shell keep track of an EGA display.

`PRINTER.SYS`, a holdover from DOS 4.x and 5, is another device driver ripe for deletion. It was part of IBM's attempt to include international printer support in DOS and is rarely needed. If you want international character support for your printer, use a good international word processor or font manager.

If you use U.S. English, you may get rid of `DISPLAY.SYS`, too; it provides international support for DOS and some applications. If your `CONFIG.SYS` doesn't include a `COUNTRY=` command or a `KEYB` command, you also may eliminate `COUNTRY.SYS` and `KEYBOARD.SYS`. Although both have a `SYS` extension, they are information files rather than device drivers.

`SETVER.EXE` is potentially more useful. As a matter of course, DOS's Setup program adds a `DEVICE=SETVER.EXE` line to your `CONFIG.SYS`. This memory-resident program tells DOS to lie and report an earlier DOS version number to certain applications. DOS creates a preliminary listing during its setup, but you may review and alter the list by typing `SETVER` at the DOS prompt. If you don't use any of the applications on the list, remove the `DEVICE=SETVER.EXE` command from your `CONFIG.SYS` file. Otherwise, remove as many applications as possible to reduce the amount of memory that `SETVER` needs.

You also should evaluate whether to bother loading `ANSI.SYS`; it can be useful but is rarely necessary. If you like to use colors on the DOS screen, or if you have an application that requires it, by all means load it with a `DEVICE=` or `DEVICEHIGH=` command. If not, remove the line and save yourself some memory.

If you have an 8088 or 80286 computer with expanded memory, you may want to use `RAMDRIVE.SYS` (`VDISK.SYS` in some DOS versions) to create a RAM disk, because it can make good use of that memory. If you have extended memory, a disk cache such as `SMARTDrive` often speeds up a system more than a RAM drive. However, you still may want to create a small RAM drive of 512K or so, especially if your applications make heavy use of temporary files.

DoubleSpace (included in MS-DOS 6.0 and 6.2, but not 6.21) is also very useful. If you selected this disk

compressor in DOS's Setup program, DOS starts it at bootup, using the invisible file `DBLSPACE.BIN`, which resides in the root directory of your host drive. All the compression capabilities and most other features associated with this disk doubler are enabled each time you boot; they don't require directives in `CONFIG.SYS` or `AUTOEXEC.BAT`. Loading `DBLSPACE.SYS` in `CONFIG.SYS` can be advantageous, however.

First of all, you may choose whether to load the driver into conventional memory using `DEVICE` or `DEVICEHIGH` to load it into upper memory. Before this command is executed, most of the DoubleSpace code sits in the high end of conventional memory, waiting to move itself down toward the lower end after `CONFIG.SYS` loads all device drivers.

In addition, `DBLSPACE.SYS` "locks down" the drive ID letters that DoubleSpace will use for drive swapping. If you don't load this pseudo device driver, DoubleSpace's choice of drive letters stays flexible until after other device drivers are loaded. You may want to load `DBLSPACE.SYS` just to take advantage of the lock down, especially if your `CONFIG.SYS` offers a menu whose setup options present a differing number of drive choices—perhaps because one configuration loads a RAM disk and another uses `DRIVER.SYS` to alter drive assignments.

After deciding whether to load DOS's various device drivers, you must take care of the device drivers accompanying your non-DOS add-ons. I prefer to load drivers such as these at the end of `CONFIG.SYS`, but this may not suit you or your system.

Optimization, the Final Step

When you're certain that your edited `CONFIG.SYS` file boots and runs properly, you're ready to use a memory optimizer, such as DOS 6.x's `MemMaker`, to load as many memory-resident programs as possible into upper memory. But first, transfer a copy of your revised `CONFIG.SYS` to your emergency bootup floppy disk.

When you run `MemMaker`, it asks you to choose an Express or Custom setup. For greater flexibility, I recommend choosing Custom. Answer the on-screen prompts, pressing F1 if you need help and choosing no when `MemMaker` asks about a program that you know can't be loaded high successfully.

After you provide the necessary preliminary information, `MemMaker` tells you to restart your system. Then `MemMaker` and its auxiliary programs, `CHKSTATE.SYS` and `SIZER.SYS`, monitor how your system allocates memory as each device driver and memory-resident program loads; later, it uses this information to determine your optimum memory setup. To achieve this, it may add new commands or alter existing ones—perhaps changing certain `DEVICE` commands to `DEVICEHIGH` and adding an `/L` or `/S` switch.

Its work done, `MemMaker` prompts you to restart your system. If all goes well during the trial start-up,

simply accept the new configuration, exit MemMaker, and type `MEM /D/P` to get a rundown on memory use. (See your manual for troubleshooting assistance.)

If you are looking for further ways to optimize, bring the `MEMMAKER.STS` file into an editor and then search for the `[SIZEDATA]` section. It lists the memory requirements (under `[MAXSIZE]`) of each driver or TSR that `CONFIG.SYS` and `AUTOEXEC.BAT` load. Using this information, reorder your `DEVICEHIGH` commands, placing the larger drivers first. (Remember, though, that

you must load `HIMEM.SYS` and `EMM386.EXE` first.)

Without a doubt, creating an efficient `CONFIG.SYS` file requires effort and thought, but the rewards are considerable. When you're done fiddling, your computer should boot more quickly and provide your DOS applications with as much conventional memory as possible. Sometimes, even a few more bytes of memory will allow that neat game (or RAM-hungry critical application) to run when it otherwise would refuse. No one I know would complain about that. ■

FLIPPING FLOPPIES WITH `DRIVPARM` AND `DRIVER.SYS`

The `DRIVPARM` command and the device driver `DRIVER.SYS` work separately or together to solve frustrating floppy drive problems.

DRIVER.SYS vs. DRIVPARM

`DRIVER.SYS` creates a new drive letter for either an internal or external floppy drive. If you have an external floppy drive, you may use `DRIVER.SYS`, instead of a third-party device driver, to create a drive letter for it. If you use `DRIVER.SYS` to create a new drive letter for an internal drive, you then may use `COPY` or `XCOPY` to copy data from one floppy disk to another in the same physical drive. DOS will prompt you to change disks as necessary. (If you have only one floppy drive, DOS lets you do this automatically.)

When you use `DRIVER.SYS`, it assigns the next available drive letter to the drive in question. For example, if you use `DRIVER.SYS` to add an external floppy to a system that has a hard disk, the letters A and B are allocated to your internal floppy drives, C is used to designate your hard disk, and D is assigned to the new, external floppy drive.

In addition to letting you use external floppy drives, `DRIVER.SYS` lets you avoid the pause that DOS requires to decide whether a drive is using a double-density or high-density disk. First, use `DRIVPARM` to define a floppy drive as high density, then use `DRIVER.SYS` to create a new drive letter and define that drive as double density. Thereafter, you may format both kinds of disks in the drive without using `FORMAT`'s `/F` switch, and you won't have to wait while DOS tries to determine the capacity of each disk you put in the drive.

Unlike `DRIVER.SYS`, `DRIVPARM` doesn't create a new drive letter. It does, however, let you establish new parameters for a floppy drive, tape drive, optical drive, and even a hard drive (although you'll rarely need to change hard drive parameters). `DRIVPARM` also lets you use a 3.5-inch drive or high-density drive in an older computer whose BIOS doesn't support such drives. Finally, you may use `DRIVPARM` to make your system work even if DOS mistakenly thinks your drives offer change-line support. Drives that have this feature are capable of detecting whether you have removed one disk and inserted another.

A Few Words on Syntax

You must invoke `DRIVER.SYS` and `DRIVPARM` in

`CONFIG.SYS`, and their syntax is similar.

For `DRIVER.SYS`, use `DEVICE=DRIVER.SYS [parameters]` or `DEVICEHIGH=DRIVER.SYS [parameters]`. For `DRIVPARM`, use `DRIVPARM=[parameters]`

The list of parameters is as follows:

`/C`—change-line support. Include this parameter if the drive can determine whether the door has been opened and the disk changed. Exclude this parameter if the drive lacks this capability. If DOS continues to display the directory of one disk after you replace it with another, the drive doesn't offer change-line support.

`/D:number`—the number of the drive. The values for drives A and B are zero and 1, respectively. For `DRIVER.SYS`, the first external floppy drive is drive 2. For `DRIVPARM`, drive 2 is drive C, the first hard disk.

`/F:n`—the drive's form factor, or type. Replace *n* with a number. Because the permissible values and their meanings vary among DOS versions, you should check your DOS documentation or on-line help to determine the correct setting.

`/H:h, /T:t, /S:s`—sets the number of heads, tracks per head, and sectors per track. Unless you are configuring an unusual device, you won't need these parameters, because the defaults determined by the form factor will be fine. But if you are installing an optical disk drive or setting up a tape drive as an auxiliary hard drive, you may need these switches.

`/I`—installs the support code. This parameter is available only in `DRIVPARM`. It tells DOS to install the necessary driver code for the disk drive. You must use this parameter if you have an old BIOS that does not include support for your drive.

`/N`—nonremovable disk. This parameter, which you may use only with `DRIVPARM`, tells DOS that the disk will never be changed. Drive access will be faster because DOS won't have to check whether the disk was changed before attempting to access the disk.

—H.B.

HOW TO STREAMLINE AUTOEXEC.BAT

Time spent refining AUTOEXEC.BAT is time well spent. Among the rewards are faster program execution, more available conventional memory, and a more colorful and informative DOS prompt.

by Hardin Brothers

WHenever clients or friends ask for help with their computers, I begin by looking at their AUTOEXEC.BAT files. My main objective isn't spotting errors but uncovering clues about the user's computing personality that will help me suggest changes in the system's start-up configuration. □ What does *your* AUTOEXEC.BAT say about *you*? Well, if it's short and to the point, I'd say you take a business-like approach to computing. If it makes liberal use of ANSI.SYS sequences,

it tells me that you're the more flamboyant type, someone who enjoys watching your PC jump through hoops. Of course, your computing style may lie somewhere in between, but no matter what type of user you are, I'm willing to bet that, with a little thought and typing, you can improve your AUTOEXEC.BAT. If you don't have an AUTOEXEC.BAT, one of the biggest improvements you can make to your computing is to create one. (See the sidebar "Editing AUTOEXEC.BAT" for advice on creating and editing this file.)

The Three Faces of AUTOEXEC.BAT

At its most basic, AUTOEXEC.BAT shields you from minor annoyances, such as time and date prompts and an uninformative DOS prompt. The requests for confirmation of the time and date are relics from the

days of 8088 computers, when systems didn't have built-in clocks. If you have an 80286 or higher computer, you shouldn't have to confirm the date and time each time you turn on and boot your system, but you'll have to if DOS can't find an AUTOEXEC.BAT. And without this important start-up file for DOS versions prior to 6.0, you'll have to make do with a DOS prompt that tells you only the current drive. You can work more efficiently if your AUTOEXEC.BAT tells DOS to display a prompt that shows the current directory's complete path.

But AUTOEXEC.BAT does more than merely eliminate annoyances; it gives you control over three very important aspects of system setup. First, it lets you load memory-resident programs (also called Terminate-and-Stay-Resident programs, or TSRs), utilities that

sit in memory and wait for a signal to spring into action. Some TSRs, including those shipped with DOS, become part of DOS and do their work unobtrusively. Many commercial and shareware TSRs “pop up” when you press a particular key or key combination. You work in the pop-up program—perhaps reviewing or recording notes, dialing phone numbers, or displaying help screens—and then press another key sequence to return to DOS or your previous application.

You also may use AUTOEXEC.BAT to create and modify your computing environment. If, for example, you don't like DOS's standard white-on-black screen, you may add colors using AUTOEXEC.BAT's PROMPT and ECHO commands. Or you may simplify the names of the subdirectories you use often and tell DOS where to look for your favorite applications.

Finally, you may use AUTOEXEC.BAT to run one or more programs. If you want to avoid the command line as much as possible, your AUTOEXEC.BAT file can take you right to a menu, a graphical user interface (such as the DOS shell or Windows), or your favorite application. You may never have to look at the bleak DOS prompt at all.

The Right Way to Load TSRs

If you're looking to improve your AUTOEXEC.BAT, one of the best places to start is with the TSRs it loads. Whether you realize it or not, your AUTOEXEC.BAT already may load several of DOS's TSRs. Some of the more notable ones are DOSKEY, a program that lets you edit the command line and create macro commands; FASTOPEN, a risky-to-use utility that speeds up certain file operations; SMARTDRV, a disk-caching program; MSCDEX, which provides CD-ROM support; POWER, a battery-power manager for laptops; UNDELETE, a file-recovery program; and VSAFE, an antiviral utility.

One mistake that users often make is running a collection of these programs in their AUTOEXEC.BAT, even if they don't use all the programs very often. If you do this, each utility grabs a chunk of your system's conventional memory. If you load enough TSRs, you may not have enough memory to run ordinary applications, such as a word processor or spreadsheet.

You should be selective not only about which programs AUTOEXEC.BAT loads, but also about where DOS loads them. Instead of placing DOS TSRs in conventional memory, use LOADHIGH (LH) to store them in Upper Memory Blocks (UMBs), out of the way of your application programs. To do this, however, you first must include two commands in your CONFIG.SYS file: DEVICE=EMM386.SYS and DOS=UMB. Also, remember that whenever you add or change a LOADHIGH command, you should rerun DOS's memory manager, MemMaker, to optimize the use of your system's UMBs. (Third-party memory managers such as

Quarterdeck's QEMM and Qualitas' 386MAX offer their own equivalents of LOADHIGH and MemMaker.)

Another way to conserve conventional memory is to install as many TSRs as possible before invoking the PROMPT, PATH, and SET commands. These commands add information to the environment, a message area in which you may leave notes to DOS. DOS passes a copy of the environment to every program you run, even your TSRs. (You may view the current contents of the environment by typing SET at the DOS prompt.)

Some TSRs, including those supplied in the latest versions of DOS, release the memory space used by their copy of the environment, but many commercial and shareware TSRs aren't as considerate. The smaller the environment passed to an inconsiderate TSR, the less memory you'll waste. Generally, the memory savings for each program are small, but those savings can mount up if you use several TSRs.

Of course, if you invoke a TSR before invoking PATH, you must supply its complete path. (For example, instead of using a command such as LH DOSKEY, you must use LH C:\DOS\DOSKEY.) This may seem inefficient, but it actually speeds execution of AUTOEXEC.BAT, because DOS won't have to search through the directories listed in PATH to find the programs it must load.

The most difficult decision you face regarding TSRs is determining which ones you should load in every computing session. You must balance convenience against memory use. The TSRs you use several times an hour are undoubtedly essential, but those you use once a day or once a week probably aren't worth the memory they consume.

I can't suggest which commercial and shareware TSRs might benefit you, but I can tell you which of DOS's TSRs are most helpful and which you should avoid (see the sidebar “DOS 6's TSRs: Which Ones Do You Need?”). Feel free to disagree with my opinions, but consider every TSR carefully before deciding whether a program's capabilities earn it a place in your computer's memory.

A Faster Disk Drive

After installing the TSRs you need, it's time to turn your attention to preparing your computing environment. I recommend that you start by improving the efficiency of the RAM disk setup in your CONFIG.SYS (see my article “Making the Most of CONFIG.SYS” on page 7). To make good use of a RAM disk, a portion of memory used as a hard disk, you should transfer COMMAND.COM and your frequently used batch files to the disk. Transferring COMMAND.COM ensures that when you exit from a program, the DOS prompt appears on screen as quickly as possible. The speedup is particularly noticeable if you have a slow hard disk, a laptop that turns off the hard disk to save battery power, or if you are using a computer that lacks a hard drive.

EDITING AUTOEXEC.BAT

When you tinker with your AUTOEXEC.BAT file, you may make some mistakes. Most of them will be harmless, and DOS will simply display an error message on your screen. But, in some cases, an error in your AUTOEXEC.BAT file will keep your computer from booting up.

Before you modify your AUTOEXEC.BAT file, make an emergency bootup disk. Format a blank disk in drive A using FORMAT's /S (system) option. Then make sure that you can boot your computer from that disk and edit the AUTOEXEC.BAT file on your hard drive. If you have an emergency bootup disk, you can recover even if a mistake in your AUTOEXEC.BAT file (or CONFIG.SYS file) locks up your computer. (If you're running the latest edition of MS-DOS, version 6.2, you may isolate problem lines by pressing F8 to execute AUTOEXEC.BAT one line at a time.)

Also, it is a good practice to make a backup copy of your existing AUTOEXEC.BAT file—naming it AUTOEXEC.BK or something similar—and then make your changes. When you're done and are sure that all of your changes work, make a backup copy of the new file so that you can undo future changes.

Making Changes

AUTOEXEC.BAT is a simple text file. You may use the Edit program, supplied with DOS 5 and later, to alter your AUTOEXEC.BAT file. You should be aware, however, that Edit doesn't automatically save a backup of the original file, as some word processors and text editors do. If you have an

earlier version of DOS, use your word processor or text editor to work on AUTOEXEC.BAT, saving the file in ASCII (text) format. If you save AUTOEXEC.BAT in the special format your word processor uses for most files, DOS will encounter problems as it executes the file. Sometimes DOS will simply report an error in such cases, but errors of this sort can lock up your computer.

As you alter your AUTOEXEC.BAT file, the most important thing you can do for your future sanity is to document each change. Because DOS ignores any line in AUTOEXEC.BAT (or any other batch file) that starts with REM, you may use REM to remove a line from AUTOEXEC.BAT temporarily or to leave comments for yourself. For example, if you decide to experiment by changing the amount of memory available to the SMARTDrive disk cache, part of your AUTOEXEC.BAT file might look like this:

```
REM Old setting:
REM C:\DOS\SMARTDRV 1024
REM Try different cache size:
SMARTDRV 512
```

REMs slow down AUTOEXEC.BAT so little that you won't notice much difference in execution time, unless you boot from a floppy disk. Even if you have hundreds of remark lines, the additional information is worth much more than the extra second or two you'll wait during each boot up.

—H.B.

Storing batch files on the RAM disk lets DOS run them without pausing to read each line from the disk.

Let's assume that your RAM disk is drive E. Your first step, then, is to use a command such as the following to copy COMMAND.COM to that drive:

```
COPY C:\COMMAND.COM E:\
```

Next, you must tell DOS where to look for COMMAND.COM by using the SET command to change the environment entry called COMSPEC:

```
SET COMSPEC=E:\COMMAND.COM
```

(Make sure you don't leave a space on either side of the equal sign.)

Then, if you normally store batch files in your C:\BATCH subdirectory and you have room for all of them on your RAM disk, add these lines to AUTOEXEC.BAT:

```
MD E:\BATCH
COPY C:\BATCH\*.BAT E:\BATCH
```

In addition, a RAM disk is a great place to put the

temporary files that DOS and application programs sometimes need. You can tell DOS, and most applications, to use your RAM disk for temporary files by using the SET command again:

```
MD E:\TEMP
SET TEMP=E:\TEMP
```

Next, you should establish the DOS search path, the list of the subdirectories you want DOS to search when you ask it to run a batch file or program. Your PATH statement should be as short as possible and should list the subdirectories with the fewest entries first. A typical setting might look like this:

```
PATH E:\BATCH;C:\DOS;C:\UTIL
```

Don't let programs add their own subdirectories to the path when you install them. Only a few poorly written programs really need to do so, and every new entry in PATH slows down your computer when you type the name of an application to run.

I already can hear some of you grumbling about having to type the full path when you want to run an

DOS 6'S TSRS: WHICH ONES DO YOU NEED?

DOS 6.x contains 16 TSRs. Some are useful, some outdated, and a few are downright dangerous. Here's how I see them:

APPEND.EXE creates a search path that lets programs find their data files. A command that had some use 12 years ago when it helped DOS 1.x programs maintain compatibility with DOS 2.x, APPEND.EXE is inherently dangerous. Don't use it, especially if you run Windows; it may cause files to disappear.

ASSIGN.EXE, another dangerous command, was relegated to the supplemental disk in DOS 6.x. It lets you fool a program into thinking that its files are on a specific disk drive. Like APPEND, it is both outdated and potentially lethal. Microsoft recommends against using ASSIGN; I agree.

DOSKEY.COM lets you recall previous commands and create powerful macro commands; both capabilities simplify life at the DOS prompt. (The featured book excerpt "Using DOSKEY Command Macros" on page 47 explains and provides examples of DOSKEY's macro feature.) But be careful not to let DOSKEY consume too much buffer space. If you use DOSKEY mostly for command-line editing, allocate a very small buffer. If you use lots of macros, set up a larger buffer. If you want much more powerful command-line editing, take a look at NDOS in Symantec's The Norton Utilities or its shareware cousin, 4DOS, from JP Software (the latter is included on this special issue's Companion Disk; see page 79 for ordering information).

DOSSHELL.EXE is a shell program, a semigraphical interface that many users prefer over the standard DOS screen. It's both powerful and useful, especially if you don't use Windows and want to shift back and forth between applications with a key-stroke or two. Whether you use the DOS prompt, DOSSHELL.EXE, or a third-party shell is a matter of preference.

FASTOPEN.EXE is a holdover from Microsoft's early attempt to speed up hard disk access. It keeps track of the physical location of often-used files so that DOS can open them more quickly. But it can be dangerous: Never leave FASTOPEN running while using DEFRAG, and never start it from a DOS prompt running under the DOS shell or Windows. If you use DoubleSpace or a disk cache such as SMARTDrive, FASTOPEN provides no real benefits, but if you use neither, it can make an old, slow hard drive seem a bit faster.

GRAFTABL.COM loads the extended character set (ASCII codes 128–255) into memory so that you may use them in CGA graphics mode. This outdated utility was relegated to the supplemental disk in DOS 6.x, in large part because few people use CGA adapters any more. If you have a CGA display, you may use the DOS 5 version of the program under DOS 6.x or purchase the DOS 6.x supplemental disk.

GRAPHICS.COM lets you press Shift-Print Screen to print a graphics screen, but it works only if you have one of the printers supported by the program. Few users need this program because most graphics programs include a way to print images or display them on screen. The advent of Windows also has helped make GRAPHICS.COM obsolete. If you run your DOS graphics programs under Windows, you can copy the screen to the Clipboard and then print it from Paintbrush or another Windows graphics program.

MIRROR.COM, DOS 5's prelude to the more powerful UNDELETE.EXE program in DOS 6.x, is outdated. If you use DOS 6.x, you'll probably want to run UNDELETE instead.

MODE runs either as a TSR or a standard application. It runs as a TSR if you use it for printer redirection (with a command such

application. You're no doubt wondering why you need to type a command such as C:\APPS\XYZ\XYZ.COM when you could adjust your PATH statement and simply type XYZ to start the same program. To my mind, the convenience of typing shorter application names isn't worth slowing down the computer, especially since you can have both simplicity and speed.

The Visual Environment

Improving aesthetics comes next. Few of us are satisfied with DOS's default text display: 25 rows of stark-white letters on a black background. You'll probably want to modify this final section of your AUTOEXEC.BAT file to create a display more to your taste.

Personally, I can find no reason to restrict my screen to 25 lines. Almost all recent computers have a VGA-or-later display system. I always use the following com-

mand to shift these computers into a 50-line display:

```
MODE CON LINES=50
```

If your printer needs one or more MODE commands to work properly, as many serial printers do, add those lines to AUTOEXEC.BAT. If you installed ANSL.SYS in your CONFIG.SYS file, you might alter the screen colors after that. The easiest way is with an editor such as DOS's Edit, which can embed escape characters in ECHO commands. The following lines set the screen color to bright white on blue, then clear the screen to make the colors appear. The left-arrow is the escape character, which Edit generates when you press Ctrl-P and then Esc:

```
ECHO ←[37;44;1m
CLS
```


as MODE LPT1=COM2) and when you use it to support international character sets. If you require either service, MODE is certainly worth the modest amount of memory it uses.

MSCDEX.EXE provides generic support for CD-ROM drives. Before you use it, you must install the CD-ROM manufacturer's device driver in your CONFIG.SYS file. You may save conventional memory by letting MSCDEX use expanded (EMS) memory for its buffers, and you can speed up your CD-ROM by increasing the number of buffers above the default value of four, especially if you don't use DOS 6.2's SMARTDrive. (If you do, be sure to start MSCDEX before SMARTDrive, so the latter program can "see" and will cache the CD-ROM drive.)

MSHERC.COM helps QBasic and Edit display graphics on a Hercules-compatible monochrome system. If you have any other kind of video adapter and monitor, you don't need it.

PRINT.EXE sends disk files to the printer while you do other work with the computer. It is seldom used because most people print from an application, rather than from the DOS prompt. You may use LOADHIGH to put the resident portion of PRINT.EXE into upper memory. If you simply want to print a document from one application while using another application, consider using Windows Print Manager (which is mediocre) or one of the third-party print spoolers for DOS and Windows.

SHARE.EXE sets up support for file sharing on a network or under a multitasking environment. You should run SHARE or LOADHIGH SHARE if you run multiple DOS applications under Windows or the DOS shell. Some applications suggest that you increase the default number of files and file locks that SHARE makes available.

Earlier versions of DOS required SHARE to support large

hard disks, but it no longer is needed for this purpose.

SMARTDRV (SMARTDrive), one of the best utilities included with DOS, is one of the few disk-caching programs smart enough to share memory with Windows when necessary. It also is smart enough to cache the host drive if you are using DoubleSpace instead of the compressed drive. The result is that the cache becomes almost twice as effective.

Although it can lose data if you turn off the computer or press the hardware reset button when a disk write is pending, SMARTDrive rarely causes problems unless you turn off your computer before you return to the DOS prompt. If you are worried about the safety of SMARTDrive, turn off disk write caching. If you have a CD-ROM drive, use the version of SMARTDrive supplied with DOS 6.2; it can cache CD-ROM data as well as hard disk and floppy disk data. (Depending on which version of DOS you own, the SMARTDRV program file will end in either .EXE or .SYS.)

UNDELETE.EXE helps you recover deleted files. Do you need it? That depends on your computing habits and how often you accidentally delete data files that you need. The safety it affords may well be worth the memory it requires. If you use it, I recommend that you use its Delete Sentry mode to provide maximum protection, unless your hard disk is almost full.

VSAFE.COM checks for virus infection while your computer is running. It has a range of checking options, which you may enable or disable when you install it. Is it worth 44K and the system slowdown that some of its options cause? Probably not, at least if you have a regular backup routine and you occasionally run MSAV or a similar virus scanner.

—H.B.

(Be sure to use a lowercase *m* in the first line.)

After you have the video display you want, you can **set up** your DOS prompt. Some people prefer a business-like prompt; others like to display folksy messages and add colors. At the very least, you should use a command such as this to include the current drive and directory in your DOS prompt:

```
PROMPT $P$G
```

You don't have to make this change in DOS 6.x, which sets up this prompt automatically.

Last but Not Least

You've made it to the final section of your AUTOEXEC.BAT, the part in which you tell DOS which program you want to run. If you want to use a shell program to insulate you from the cold, harsh DOS prompt or you

want to run a particular program, this is the place to add the appropriate command.

When all three sections of AUTOEXEC.BAT are arranged to your liking, you need to test the file. Save it, exit from the editor you've been using, then reboot your computer. If you made extensive changes, one or two errors may show up. That's normal. Repeat the edit-save-test cycle until everything works satisfactorily.

Your AUTOEXEC.BAT file should be a reflection of your computing personality; as such, it should make your work as pleasant as possible. If your start-up settings aren't quite right, keep modifying them until DOS does what you want. Even then, don't be afraid to make further revisions. Treat AUTOEXEC.BAT like a work-in-progress; as your computing habits change, it should change along with them. ■

MAXIMUM MEMORY, Minimum Fuss

It's a no-brainer: DOS 6.x's MemMaker utility will find more of that precious RAM space for you. Still, it helps to know your way around UMBs, HMA, and all the other flavors of memory.

by Dan Gookin

SOONER OR LATER THE MEMORY MANAGEMENT MESS WILL COME TO roost in your PC. Its approach can be subtle. First are a few "insufficient memory" errors, often missed and easily ignored. Then "XMS memory required" or "HMA unavailable" or "EMS memory unavailable." That's when memory management and its associated terminology mixes into an impotable brew that no PC user can swallow. Time to yelp for help. □ Fortunately for today's DOS user times aren't as desperate

as they were five years ago. The solution is to run a good memory optimizer, such as MS-DOS 6.x's MemMaker, and be done with it. That's what I did on my PC a while back, and I haven't messed with it since.

DOS Solutions

The bottom line for memory management is the same as always: Get more conventional—640K-and-under—memory available. Additionally, memory management also must harness and govern all the memory in your PC, giving it out to programs that need it. This usually is handled in five steps in CONFIG.SYS and AUTOEXEC.BAT, listed below and in the Figure.

1. Load HIMEM.SYS and let it control all of extended (1MB-and-above) memory.

2. Run EMM386.EXE to provide Upper Memory Blocks (UMBs) and expanded memory. Upper memory is a portion of the PC's memory just above conventional memory; expanded is another type of extra memory that DOS can't access directly.

3. Use the DOS=HIGH,UMB directive to load DOS high and to tell DOS to manage space allocations for the UMBs.

4. Use DEVICEHIGH= in CONFIG.SYS to load device drivers high.

5. Use LOADHIGH= in AUTOEXEC.BAT to load memory resident programs high.

Admittedly, these steps are more than anyone should have to deal with. Though third-party memory managers such as Quarterdeck's QEMM and

Qualitas' 386MAX for years have offered definite solutions to DOS's memory limitations, they still leave it partly up to you to figure out how to carry out the above steps. This is senseless because, after all, a computer should be able to do anything with a minimum of human involvement. So Microsoft solved the problem with DOS 6.x and its MemMaker utility.

Using MemMaker

To effectively manage your PC's memory today you need do only two things: Buy MS-DOS 6.0 or 6.2 and run its MemMaker utility. MemMaker does it all, setting up the proper commands in CONFIG.SYS and AUTOEXEC.BAT for you automatically.

You start MemMaker by typing its name at the DOS prompt:

```
C:\> MEMMAKER
```

The quickest way to get things done is to add the /BATCH switch:

```
C:\> MEMMAKER /BATCH
```

This directs MemMaker to configure your PC's memory as it sees best, which is just fine for most of us. The program ends with a summary screen that explains how your PC's memory situation has improved. It's impressive, but trivially easy.

In the olden days of DOS, I used to direct people in

a detailed discourse on memory management. Today, that's really unnecessary, thanks to programs such as MemMaker. But if you're still using an older DOS, take a look at the sidebar "Under the Hood with CONFIG.SYS and AUTOEXEC.BAT."

Third-Party Solutions

Nothing sold DOS users on third-party memory managers better than DOS 5, which came with the new MEM utility that did a better job than before of telling you how much memory you had, but then offered no fix. The best-known third-party memory managers, such as QEMM and 386MAX, which came with their own memory-optimization routines, provided instant relief to desperate DOS users. But since DOS now comes with MemMaker, are third-party memory managers still necessary?

The answer is truly, "no." You don't need third-party memory managers if you have DOS 6.x. While they will optimize memory more efficiently and maybe give you one or two kilobytes of extra memory, most DOS users don't need them.

If you're a memory nut, or someone who's so deeply into this that every last byte means something, then go for it. QEMM is my recommendation for technical users; 386MAX is good for the "don't want to mess with it" types. And keep in mind that these are still viable solutions for pre-DOS 6.x users. I have one DOS 5 system still lingering around that uses 386MAX simply because DOS 5 couldn't hack it.

CONFIG.SYS

- 1 → DEVICE=C:\DOS\HIMEM.SYS
- 2 → DEVICE=C:\DOS\EMM386.EXE NOEMS
- 3 → DOS=HIGH,UMB
STACKS=0,0
FILES=50
BUFFERS=32
LASTDRIVE=Z
- 4 → DEVICEHIGH=C:\DOS\RAMDRIVE.SYS 1048 /E
DEVICEHIGH=C:\DOS\ANSI.SYS

AUTOEXEC.BAT

- 5 → @ECHO OFF
PATH=C:\DOS;C:\UTIL
PROMPT \$P\$G
LOADHIGH C:\MOUSE /Y
LOADHIGH C:\DOS\DOSKEY /INSERT

Figure. The CONFIG.SYS and AUTOEXEC.BAT files above contain all five steps for DOS memory management, which are labeled 1-5. 1. HIMEM.SYS is run first, DOS's main memory-management control program. It sets up the Extended Memory Specification (XMS) on the computer, controlling all of extended memory. It also creates the High Memory Area (HMA). 2. EMM386.EXE is run to provide Upper Memory Blocks (UMBs) and any expanded memory for those programs that request it. 3. The so-called DOS configuration does two things. First, DOS=HIGH tells DOS to put itself into the HMA, saving some 43K of conventional memory. Second, DOS=UMB tells DOS that it can load device drivers and memory-resident programs into the upper memory area, into blocks of memory created there by the EMM386.EXE command. 4. The DEVICEHIGH command loads device drivers into UMBs. 5. The LOADHIGH command in AUTOEXEC.BAT loads memory-resident programs into UMBs. (All of these steps are automated by MemMaker, which also may add optional switches after the EMM386.EXE, DEVICEHIGH, and LOADHIGH commands, further optimizing the use of memory.)

UNDER THE HOOD WITH CONFIG.SYS AND AUTOEXEC.BAT

If you own MS-DOS 5 or earlier, you're missing out on the neat memory services of DOS 6.x's MemMaker. But you can make some of the changes manually that MemMaker automatically performs if you dive into editing your PC's configuration files, CONFIG.SYS and AUTOEXEC.BAT.

As a rule, though, you shouldn't do anything to AUTOEXEC.BAT after a memory manager, or an application's setup program, has made its changes. The same more or less holds true for CONFIG.SYS, but there are more opportunities for tweaking. (For guidance, look at the Figure in the main article. It shows you sample CONFIG.SYS and AUTOEXEC.BAT files that have been configured for optimal memory management.)

Here are some general recommendations for making memory-related improvements in CONFIG.SYS:

- Load DOS high using the DOS=HIGH command. This command relocates DOS itself to the High Memory Area (HMA), a special area created by the HIMEM.SYS driver in the bottom 64K of extended memory.
- Run the latest version of EMM386.EXE (unless you're using a third-party memory manager). Specify NOEMS if you don't

need expanded memory; otherwise specify the RAM option.

- Enable DOS to manage the upper memory blocks by using DOS=UMB command.
- Load device drivers into upper memory with the DEVICEHIGH= command.
- If memory space is precious, dispense with a RAM drive by removing the line that says DEVICE=pathRAMDRIVE.SYS or DEVICEHIGH=pathRAMDRIVE.SYS.
- Set up a disk cache on your system, such as MS-DOS's SMARTDrive (explained on page 30). DOS's setup program installs SMARTDrive for you. If you use a disk cache, you don't need a BUFFERS= command in CONFIG.SYS.
- Set the FILES= command to something low, usually 20. Only if you get an "insufficient file handles" type of error should you make the value larger.
- Do not disturb any changes made to CONFIG.SYS by a memory-optimizing utility.
- Set the STACKS command to STACKS=0,0 to save a few kilobytes of memory.

—D.G.

Personal Suggestions and Tips

There are about 80 million DOS computers in the world. And out of those millions of PCs there are millions of users, each with his or her own set of device drivers, memory-resident programs, expansion cards, and purposes for his or her computers. Therefore, it would be sheer madness to concoct an all-in-one memory-management solution for everyone.

So, instead of dwelling on insanity, just keep these

recommendations in mind: You should always pack your PC with as much memory as you can afford (or as much as will fit inside); use DOS 6.x's MemMaker or, if you don't have it, a third-party memory manager; and keep the emphasis on freeing up conventional memory.

Finally, don't lose perspective. Memory management is an ugly necessity of using a DOS computer, but not one worth fretting over. ■

EXPANDED, EXTENDED, EXASPERATED: MEMORY TYPES EXPLAINED

CONVENTIONAL MEMORY (also known as main memory or sometimes DOS memory.): DOS's main memory, where all of your programs run. It is limited to 640K. Since most new PCs come with 640K of conventional memory, the real issue is how to make more of it available. This is done by moving some hoggy programs out into upper memory.

EXPANDED MEMORY: Special memory used by programs that crave more than the puny 640K conventional memory. It must be added to an old 8088 or 80286 computer with a special expansion card. Then EMS software is required to make it work. On 386-and-later microprocessors, expanded memory is created using spare extended memory. This magic is accomplished with DOS 5 and 6.x using EMM386.EXE.

EXTENDED MEMORY: Memory in your 286-or-higher computer that lives way up there beyond 1MB. It is used by special advanced programs, such as Windows, that crave it and want

lots of it. In DOS, RAM drives and the SMARTDrive disk cache also are created in extended memory.

HIGH MEMORY AREA (HMA): A special part of extended memory used by DOS to store some interesting things, primarily itself and some of its programs. The HIMEM.SYS device driver creates this 64K chunk of memory from the first 64K of extended memory. Windows also uses it.

UPPER MEMORY (also known as reserved memory, upper DOS memory, or high DOS memory): A portion of the PC's memory that lives just above conventional memory. Most of it goes unused. Memory managers "fill it up" with usable memory, which creates the famous Upper Memory Blocks (UMBs). Device drivers and TSRs can be shoved up there with a memory manager, or the DEVICEHIGH and LOADHIGH commands—governed, of course, by an optimizer like MemMaker.

—D.G.

RUNNING IN THE Fast Lane

More efficient disk operations mean faster overall system performance. These four easy-to-use DOS commands make it happen.

by John M. Goodman

DISK DRIVES ARE AN INDISPENSABLE PC COMPONENT. AFTER ALL, without them you wouldn't have anywhere to keep information when you turned off your computer. But disk drives also are a drag; even the fastest ones don't begin to match the speed of your system's Central Processing Unit (CPU). □ PCs cope with this disparity by letting the CPU work only with a program or data that is in memory. Each time the CPU needs new instructions or data, it must

read information from disk. So, all day long your PC sloshes data from disk to memory and back again. Often, the speed with which it conducts these data transfers is what limits the overall speed of your computer.

Fortunately, you can pep up the inevitable transactions between your computer's CPU and its disk drives in several ways: by minimizing the number of data transfers your PC makes, by transferring data more quickly, and by doing what you wish to do yet transferring fewer bytes of data.

A number of DOS commands can help you achieve these goals, but I'll concentrate on the four that I find most effective: **BUFFERS**, **SMARTDRV** (SMARTDrive), **DEFRAG**, and **DBLSPACE** (DoubleSpace). Another command, **FASTOPEN**, sounds promising, but it can wreak

havoc with your data, so I don't advise that you use it.

BUFFERS and **SMARTDrive** were designed to help DOS get its work done with fewer disk accesses. **DEFRAG** attempts to decrease the time DOS must wait for disk accesses to be completed. In contrast, the disk-compression program **DoubleSpace** usually isn't thought of as a command that can speed up your PC. But a little-known side effect of squeezing more information on a disk is that it sometimes speeds up disk accesses.

BUFFERS

All PCs equipped with drives share one basic strategy for speeding up the transfer of data to and from a disk drive. They refuse to read or write one byte of information at a time, instead insisting on reading and writing blocks of 512 bytes. (This is the size of each of

the *sectors* on your PC's disks. From the point of view of the drive, this is the minimum unit into which your data can be broken.)

If you're running a program that wants a particular byte from a file, it asks DOS for that byte. DOS can fulfill this request only by first asking the drive to read, in one step, the necessary byte and the other 511 bytes in that disk sector. The 512 bytes are stored in a DOS *disk buffer* temporarily. Later, if the program asks for the next byte and the one after that, DOS can supply them from the buffer, instead of waiting the horribly long time (electronically speaking) that it takes to read the bytes from disk.

Visualize a DOS disk buffer as a staging area or loading dock. Programs accumulate, in a buffer area, the bytes destined for a drive. When the buffer is full, all the bytes are sent to the disk in one batch. Similarly, when the drive delivers data, it dumps a sector's worth of data into a buffer. A program then can sift through the pile of bytes and retrieve just the ones it wants.

You must have at least one DOS disk buffer, and you may not have more than 99. You specify how many you want by including a command in the following format in your CONFIG.SYS file:

BUFFERS=*nn*

where *nn* is the number of buffers you want. If you don't tell DOS how many buffers you want, it uses its default setting. Under DOS 4.0 and later, this setting, which ranges from 2 to 15, varies according to the amount of available memory and the capacity of your disk drives.

What is the "right" number? That depends, mostly on whether you use some form of disk caching, such as that offered by DOS's SMARTDrive. With disk caching, which I strongly recommend, you need very few DOS disk buffers—perhaps as few as three. You may use many more without consuming any more of your valuable conventional memory, but only if you run DOS 5 or later and you include the command DOS=HIGH in your CONFIG.SYS.

If you don't use disk caching, you must specify a larger number of DOS disk buffers; exactly how many depends on the kinds of programs you use. Often, more is better, because when you have multiple buffers, DOS doesn't overwrite the contents of one buffer until it uses all the others. This improves the odds that when a program needs to reread information it read previously (or when it needs information stored in the same sector as data read *previously*), the required data is in a disk buffer and won't have to be read from disk.

It isn't always wonderful to have lots of disk buffers, however. In fact, it can hurt you to specify too many of them, because each one takes up a little more than a half-kilobyte of precious RAM.

Most DOS versions put all the DOS disk buffers in low memory (the area between zero and the 640K mark); DOS 5 and later can put them in the High Memory Area (HMA) located between 1024K and 1088K, but only if you specify DOS=HIGH in your CONFIG.SYS file. DR DOS also lets you put its disk buffers into upper memory, the area between 640K and 1024K. But no matter where the buffers reside, they take up room that might better be used for something else.

You can easily find out just how much memory you are using for disk buffers by typing MEM /D at the DOS prompt. Look for the line or lines that mention the word BUFFERS. (Remember, if you have DOS 5 or earlier, the size of an area of memory is given in hexadecimal. In that numbering



DON'T FORGET THE ALSO-RANS

BUFFERS, SMARTDrive, DEFRAG, and DoubleSpace aren't the only DOS commands that affect disks and drives. Because these "other" commands don't directly affect the speed with which you can access data on your disks, I didn't discuss most of them at length in the main article. Two, however, are worth mentioning: the old standby CHKDSK, and ScanDisk, which is new in MS-DOS 6.2. These commands help ensure that your disk's data is stored safely and that the pointers DOS uses to find the data are properly organized.

Use CHKDSK (or ScanDisk if you have it) periodically to find and fix any "mess-ups" that may occur on your disk. This is important not so much because it improves the speed with which your disk drives work, but because it ensures that your disks work at all. There's little point in having your disks work quickly if they give you nonsense instead of the programs or data you want.

If you use CHKDSK, first run it without any switches (a drive letter followed by a colon is OK, though). If it reveals that problems exist, and if they are limited to chains of lost clusters, it's safe to run CHKDSK /F and let it convert the lost chains to files or simply discard them. (If you convert them to files, look at each one with a file-viewing program just in case one of them is a file you lost and want to keep. If not, then delete it.)

If CHKDSK says you have more serious problems, you may well want to find a better program than CHKDSK to fix those problems. ScanDisk is somewhat better, but one of the good third-party utility programs, such as Symantec's The Norton Utilities, may be what you need at that point.

—J.M.G.

scheme, 200 is equivalent to 512 decimal.)

What's the bottom line? In most cases, you may safely set `BUFFERS` to 14. If you use a disk-caching program, you shouldn't need more than this. If you also include the command `DOS=HIGH` in `CONFIG.SYS`, DOS attempts to put the buffers in an unused portion of the HMA.

The only time you should set up fewer buffers is if you want to use the HMA for something other than DOS. People who run Quarterdeck Office Systems' DESQView, for example, sometimes find it better to let that program have the HMA to itself. In cases such as these, you should set `BUFFERS` to three.

You'll need more buffers only if you don't use disk caching. But then, if you don't, you aren't going to get optimal disk performance anyway.

SMARTDrive

Savvy PC users have employed disk caching for years. Some disk caching is done by special hardware on a special kind of disk controller. A cheaper, easier approach, and one that is almost as effective, is to use a software cache.

A disk cache is merely an extension of the notion of a disk buffer. Simply put, a cache is a region of RAM used to store all the information that was read from or written to a disk recently, along with the location on disk where the data normally lives.

Whenever a program wants to read from a location it read previously, the disk cache supplies the information from the cache—at electronic speed. When a program wants to write information to the disk, the cache program checks whether that exact information already was written to the specified location on disk. If so, it doesn't bother to write it again. Either way, the program gets back to work much more rapidly than otherwise would be the case.

The principal advantage of a disk cache over DOS's disk buffers is that because it can use expanded or extended memory, it can keep a lot more information in RAM. In addition, once it fills up all allotted RAM, a well-designed disk cache is cleverer than DOS about deciding which information to keep around for reuse.

Note that the type of disk cache described above—a *read- and write-through cache*—can't do anything to speed up the first reading of information from disk, nor can it speed up the first write-back of information. Only subsequent reads and writes are quicker. Fortunately, however, PCs do a lot of repetitive reading. Writing is less repetitive.

A *deferred-write* cache strategy offers further speed advantages. Like a read- and write-through cache, a *deferred-write* cache accepts information from a program, along with the location on the disk to which that information will be written. But, as its name suggests, a deferred-write cache postpones writes,

telling the program that the data was written to disk safely even though it wasn't. The program goes back to doing its job with almost no delay. Later, when the PC is idle, the cache program takes control of the PC and sends the data in its cache memory to the proper disk. Thus, a deferred-write cache effectively speeds up even the first time information is written to any location on the disk drive.

DOS versions 4 and later include a caching program called SMARTDrive (`SMARTDRV.SYS` or `SMARTDRV.EXE`), but until DOS 6.0 was released, the program was mediocre. Most knowledgeable users ignored it, using a third-party disk cache instead. SMARTDrive now has caught up to its competitors and has the distinct advantage of not costing you anything extra.

Watch out, though! In version 6.0, SMARTDrive performs deferred-write caching by default. This has caught many a user by surprise. You'll lose data if you turn off or reset your computer at the wrong time, before it has finished writing something to the disk. Depending on what you were doing, the result could range from annoying to tragic.

Both PC DOS 6.1 and MS-DOS 6.2 include features that supposedly protect you from losing data. In both DOS versions, SMARTDrive defaults to doing only read- and write-through caching. Also, even if you turn on deferred-write caching, SMARTDrive ordinarily prevents `COMMAND.COM` from displaying a DOS prompt until all of the information in the cache is written to disk. This can keep you from thinking that your program has finished its work when it really hasn't.

If you have the discipline to wait until the disk cache is flushed before removing a floppy disk from the drive or turning off or rebooting your PC, then you'll enjoy the speedup you can obtain by enabling deferred-write caching. (For instructions on using SMARTDrive, see the feature "Tuning Up SMARTDrive" on page 30.)

DEFRAG

The DEFRAG file defragmenter is another DOS 6.x tool that can help you speed up your disk accesses—if you use it periodically.

Two factors control how quickly you get information from a disk drive. The first, *access time*, measures how long it takes to position a drive's read head over the start of the information. The second, *data-transfer rate*, measures how rapidly the information can flow off the disk once it is located. Usually access time isn't the most important number, because although it may take 0.03 seconds (30 milliseconds) to get to a file, it may take several seconds to read the entire file into memory.

File fragmentation can upset that balance. If you

have numerous badly fragmented files (files in which the data isn't stored in consecutive disk sectors), you must endure an access-time delay whenever DOS must find and read another file fragment. When a disk is badly fragmented, it can easily take twice as long to read a file.

Fortunately, the solution is simple. Just run `DEFRAG` and let it move your data around on the disk until all the files are stored in contiguous clusters. Typing `DEFRAG C: /U` tells DOS to defragment only files, leaving empty spaces between files. Typing `DEFRAG C: /F` asks DOS to defragment the files and the free space. The latter procedure takes a bit longer but makes subsequent file fragmentation build up more slowly.

In addition, you may specify a sort order, using the switch `/S[:x]` (the colon is optional). You may replace `x` with `N`, `E`, `D`, or `S` to sort files in ascending order by name, extension, date, or size (respectively). If you follow the letter that indicates sort order with a hyphen (-), it tells `DEFRAG` to sort in descending order. This command, for instance, places the files on the disk in ascending order by filename extensions and makes sure all free space is in one large clump at the end:

```
DEFRAG C: /F /S:E
```

To accomplish the same thing but put the files in descending order, the command is:

```
DEFRAG C: /F /S:E-
```

If you simply type `DEFRAG` on the command line, DOS starts a full-screen interactive defragmenting session. You then specify the drive, method, and sort order by choosing entries from menus.

Third-party disk defragmenters often include many more options, such as the ability to move the directory files to the front of the disk and the option of sorting the locations of the files according to their filenames or extensions or according to which directory files contains their names. These options help you get the best possible performance from your drive, but, for most people, the speedup these additional steps provide is unnoticeable. If you don't want to buy a third-party utility but really want that kind of optimization for your disk drives, and you're willing to work at getting it, see the sidebar "Defragmenting to the Max."

DoubleSpace

Finally comes the real oddball in this article's assortment of DOS utilities, DoubleSpace (its filename is `DBLSPACE.EXE`).

I call it an oddball because, although you undoubtedly have heard of this command and know that it

can effectively double the capacity of your hard disk, almost no one talks about what it can do for the speed of your PC. (The major DOS alternatives offer equivalent disk compressors. In DR DOS 6 it's called SuperStor; PC DOS 6.1 includes a special version of SuperStor called SuperStor/DS. Novell DOS 7, the successor to DR DOS, comes with a special version of Stacker 3.1 for Windows and DOS.)

Remember, minimizing the number of disk accesses is only one way of speeding up your PC's disks. Another is simply reading and writing fewer bytes to disk. If you use an on-the-fly data compressor to remove the redundancy in the data to be written, and that compressor works quickly enough, you'll speed up your PC while making your disk drives look as if they are roomier than they are.

All data compressors have the same goal: increasing the amount of information you can store on your disk. But such programs speed up things only if your PC's CPU is fast enough, or if you use a special data-compression board. (At present, Stacker [from Stac Electronics] is the only data-compression product on the market that can use a hardware compression board.)

How fast a PC do you need? My informal tests suggest the answer is that any PC with a 20MHz 386SX or faster CPU can reduce disk-access time. The speed of disk reads increases more than the speed of disk writes. Reading and writing files containing little redundant data, such as ZIP or GIF files, slows down your system slightly.

If you have a slow PC and don't have a data-compression board, however, using on-the-fly data compression slows down your machine's overall performance. Although you still may wish to use DoubleSpace on such a system, because it lets you store more information on your disks, you should realize that you won't get a speedup in every case.

Because the compression performance you want, rather than speed considerations, dictates which setting you should use with DoubleSpace, I won't discuss the command's various options here. You can find syntax advice in any good manual for DOS 6.0 and later, and on the "DOS Reference Card" that comes with this magazine. Remember, the real reason to use on-the-fly data compression is to get the apparent increase in disk capacity. It has an effect on speed, but in most cases, the effect is minimal.

(At this writing, Microsoft was recalling all store copies of MS-DOS 6.2 after losing a lawsuit brought against it by Stac Electronics. The replacement upgrade, MS-DOS 6.21, does not contain DoubleSpace).

You Can Get There from Here

If you're like most PC users, you want your PC to work as quickly as it can. Because drives often are the

biggest cause of system slowdowns, streamlining disk operations usually is the most effective means of improving a computer's speed. Boosting drive speed is simple: carefully edit your CONFIG.SYS and AUTO-EXEC.BAT files, run a few DOS utilities, and you're home free.

For a vast majority of people, the most important steps are establishing a modest BUFFERS setting (14 is good starting point) and running an efficient disk-

caching program (DOS's SMARTDrive and its defaults will work fine for most users). Running DoubleSpace or another on-the-fly disk compressor is more important for freeing up disk space than for speeding up your PC, although it has the ability to do both. Running DEFRAG occasionally will help your PC continue performing at its best. All of these strategies have a common aim: getting more work done in less time. ■

DEFRAGMENTING TO THE MAX

The files containing your programs and data aren't the only information stored on your disk. DOS needs pointers to help it find those files. These pointers are, in part, stored in special files called directories.

If you have more than a few hundred files on your disk, you need to organize them into directories, and you may further organize them into subdirectories and sub-subdirectories. Even if you only have a few files (as you may on a floppy disk), this can be a useful strategy.

Before DOS can read information from or write information to a file on the disk, it first must find out where that file is located. This means that it must read the root directory to learn where it can find the directory file containing the pointers to the file. After it reads that directory file, it then can read the file for which you originally asked. If your disk contains subdirectories and sub-subdirectories, DOS requires even more steps to manipulate the data in your file.

One way to speed up access to your files is keeping all the directory files near the beginning of the disk. The root directory always resides there, but the other directory files could end up almost anywhere. And, like your data files, they can get fragmented.

Certain third-party disk "defraggers" let you defragment the directory files and place them at the beginning of the disk, right after the root directory. Unfortunately, DOS's DEFRAG doesn't have this option. If you're willing to go through the following steps, however, you can achieve the same effect without investing in extra software:

1. Back up all the files on the fragmented drive (a logical volume such as drive E). Check that your backup is a good one by restoring at least a random sampling of those files to another drive letter and then checking the resulting files against the originals with DOS's COMP or FC command.
2. Log onto the drive volume in question (by typing E:, for example). Type TREE > PRN or TREE > A:\filename. Putting the file on paper or a floppy disk ensures that you can get at this information even after you take the next step.
3. Type FORMAT x:, where x is the letter of the drive you are

optimizing. Answer the questions it poses and confirm that you want to wipe out all data on this drive.

4. Surface-test the volume. This step is optional, but it's a very good idea, because you really don't want to attempt to store information on an area of the disk that is damaged. Contrary to what many people believe, FORMAT does little or nothing in this regard. Currently, the best surface scanner for DOS disks is Gibson Research's SpinRite III. MS-DOS 6.2 includes ScanDisk, which is better than nothing, but it doesn't do nearly as good a job as SpinRite does.

5. Using the output of the TREE command, employ CHDIR (CD) and MKDIR (MD) to reconstruct its subdirectory structure. Of course, if you're interested in developing the most efficient subdirectory system, you can revise your directory tree.

Start by creating the lowest level subdirectories that your programs access most frequently. In each of these subdirectories, begin by creating the sub-subdirectories accessed most frequently.

6. Restore your files to this logical volume. Doing this after the directories are created ensures that all the directory files are placed on the disk before any of the other files. Because the disk was freshly formatted, each backup file you restore is placed on the disk in a contiguous set of clusters.

You can fine-tune this further. First, restore only the executable files, those with extensions such as COM, DLL, EXE, and OVL. Then, restore your data files and any others you backed up. This puts the files that don't change (and therefore don't become fragmented over time) in front of those that do change.

This procedure will fail to give you an optimally organized disk only if you have a huge number of files in a directory. In that case, as you're restoring files to that directory, DOS will need to extend the directory file past the first cluster. At that time, it will fragment that particular directory file. Unless you frequently access files near the end of that directory file, this won't slow down your PC noticeably.

—J.M.G.

TUNING UP SMARTDrive

Tweaking DOS's disk-cache software can spark your PC's performance by some 200 percent.

by Stan Miastkowski

SMARTDRIVE IS ONE OF DOS'S HIDDEN GEMS. A DISK-CACHING UTILITY that's set up automatically when you install DOS, SMARTDrive can make a dramatic difference in how fast your PC runs. □ The program (look for SMARTDRV.EXE in your AUTOEXEC.BAT file) speeds up your system by using part of your PC's RAM as a storage place for frequently accessed data. □ Without caching, every time you start a program or call up a file, your PC must go out to the hard disk to get the files you

need. The process is fast, but a hard disk is only a mechanical device; the read/write head must move to the data's location before reading it. If your data is sitting in system RAM, however, for all practical purposes, access is instantaneous.

A disk cache is even faster and more versatile than a RAM disk (created by DOS's RAMDRIVE.SYS), which stores in RAM data that you have copied over manually. SMARTDrive's cache is dynamic; the program continually changes the contents of the cache by watching how you work, keeping track of files you use most, and essentially making educated guesses about which files you'll need next.

Historically Speaking

Caching isn't a new technology. In fact, it was devel-

oped many decades ago for mainframes. And SMARTDrive itself has gone through many changes over the years.

Through DOS 5.02, it was a simple device driver (SMARTDRV.SYS) that installed in your CONFIG.SYS file, and it had no options other than setting the size of the cache and choosing whether to put the cached data into extended or expanded memory. A SMARTDrive revolution of sorts began with the release of Windows 3.1. This Windows version installs a new SMARTDrive, which replaces the old device driver with a utility (SMARTDRV.EXE) that runs from your AUTOEXEC.BAT and offers a raft of fine-tuning options.

With the release of MS-DOS 6.0, Microsoft further changed the program by adding write caching to SMARTDRV's read-caching capability. Write caching

makes a noticeable improvement in PC performance. Normally, every time a program must write some information to the disk, a perceptible delay occurs while the hard disk does its thing. With write caching, the program stores data to the RAM cache instead of immediately writing it to the hard disk. This means you can return quickly to your work. The caching program then writes the data to disk during free processor cycles, and always within a few seconds.

The addition of write caching proved to be so controversial, however, that Microsoft released an interim version (available on CompuServe and other on-line systems) that let users turn off write caching. (For more on this, see the section titled "Write or Wrong?".)

The current state of the caching art—at least from Microsoft's viewpoint—is SMARTDrive 5.0, bundled with DOS 6.2x. It features a wide range of tuning options and is the version I'll concentrate on here. Besides caching your garden-variety hard and floppy disks, it also caches CD-ROM drives as well as compressed disk volumes created by DOS 6.20's DoubleSpace utility.

Stashing the Cache

SMARTDrive has become so useful, simply because RAM has become comparatively inexpensive over the past couple of years. Systems with 4-, 8-, or even 16MB of RAM aren't unusual anymore. And some of that extra RAM is the perfect place to put a cache.

SMARTDrive works best in most PCs when it is set to use extended memory for its storage of frequently used data. A typical PC has 640K of conventional memory (RAM). The rest of the first megabyte is used for video RAM and some ROMs, and in some cases, for some more RAM in which programs may be stored, called Upper Memory Blocks (UMBs). Extended memory is any you have in your PC at addresses beyond 1MB. The latest version of SMARTDrive can only use extended memory for its data storage, and then only if you have loaded an extended memory manager (XMM) such as HIMEM.SYS to turn your extended memory into XMS memory. This is the way the vast majority of PCs are now configured.

In addition to the XMM, you may be

loading a memory manager that can make some simulated expanded (EMS) memory out of XMS memory. The DOS program to do this is EMM386. (That same program also can be used to create UMBs.) SMARTDrive will only use the XMS memory as itself, and not as simulated EMS memory. For more details on these different kinds of memory and how they can be managed, see the article "Maximum Memory, Minimum Fuss" on page 22.

The SMARTDrive program itself is relatively small, requiring about 28K of conventional or upper memory to do its work, plus the extended memory used for the cache itself. By default, SMARTDrive loads itself high, using the UMBs between DOS's base memory of 640K and 1MB—assuming you're using DOS's HIMEM.SYS and EMM386.EXE device drivers to create those UMBs at start-up. (The memory feature on page 22 tells you how to do that.) SMARTDrive can load itself into upper memory without the help of the

WHICH SMARTDRIVE VERSION DO YOU HAVE?

SMARTDrive, DOS's built-in disk-caching utility, has gone through more upgrades than DOS itself. Each has offered better performance and more options. To get the most from the program, you should have the latest version, which is shipped with DOS 6.2x and works only with DOS 6.0 and 6.2x.

If you're not sure which version you have, here's a list of the program's recent incarnations. But first, find the SMARTDRV.EXE (or SMARTDRV.SYS) file, which is usually in your \DOS directory. Alternatively, look in your \WINDOWS or C:\ directory. Check the file date with the chart below. You'll also see the version number briefly flash by on your screen when SMARTDrive loads from your CONFIG.SYS or AUTOEXEC.BAT files.

Remember, if you have multiple versions of SMARTDRV.EXE or SMARTDRV.SYS on your hard disk, make sure CONFIG.SYS or AUTOEXEC.BAT refers to the latest one.

VERSION 3.13

(Shipped with MS-DOS 5)
SMARTDRV.SYS
File date: 4/9/91

VERSION 4.0

(Shipped with Windows 3.1)
SMARTDRV.EXE
File date: 3/1/92

VERSION 4.0

(Shipped with MS-DOS 6.0)
SMARTDRV.EXE
File date: 3/10/93

VERSION 4.1

(Update available through
bulletin board systems)
SMARTDRV.EXE
File date: 6/9/93

VERSION 5.0

(Shipped with MS-DOS 6.2)
SMARTDRV.EXE
File date: 9/30/93

—S.M.

LOADHIGH command. If you don't want it added into upper memory, you will need to use the /L command-line parameter to force it to go into conventional memory.

In some cases, during its installation process DOS adds the /L option to the SMARTDrive line, like this:

```
C:\DOS\SMARTDRV.EXE /L /X
```

This uses up 28K of your main memory, but it is necessary if the UMBs have been set aside for other programs. Normally DOS's installation program is smart enough to tell if SMARTDrive needs to be loaded into base memory. If it does make a mistake, don't worry; the program provides an error message saying it can't load into high memory. Then you'll have to add the /L option to the SMARTDrive line in your AUTOEXEC.BAT file.

Write or Wrong?

When you install DOS 6.2x, SMARTDrive 5.0 is placed in your AUTOEXEC.BAT file. The default line looks like this:

```
C:\DOS\SMARTDRV.EXE /X
```

The /X indicates that the program's write caching is disabled, and it is the default command. If you remove the /X from the command line, SMARTDrive's write caching is enabled, and you'll get a noticeable increase in your PC's performance. (Using drive letters in the SMARTDrive command line, along with plus and minus signs, further controls read and write caching for specific drives. See the section titled "Tuning Options.")

As I mentioned, write caching remains a controversial issue. If a power failure occurs or if you reach over and turn off or press the reset button on your PC between the time you save a file and when it's written to disk, your data is lost.

Should you use write caching? If you want maximum SMARTDrive performance, the answer is yes. I've used it extensively without problems. If you

would rather be on the safe side, however, leave /X in the program's command line. Here's one more safeguard to note: When write caching is turned on, SMARTDrive clears the cache (a process known as "flushing") when you use Ctrl-Alt-Del to reset your system.

Measuring for a Perfect Fit

As you would expect, the bigger the cache, the better a performance improvement—up to a point. Unless you tell it differently, SMARTDrive uses a default cache size that depends on how much extended memory is installed in your PC. The default values are described in Figure 1.

By default, SMARTDrive won't create a cache larger than 2MB. And caches larger than that offer only a very small performance gain, mainly as a result of how DOS works with the processor and RAM. A 2MB cache is optimal, but, on the other end of the spectrum, a cache smaller than 256K isn't much better than having no cache at all. The *bottom line*: Make your cache as large as possible, up to the 2MB limit. (If you'd like a rough measure of the relative performance gains of different settings, try HITS62, a shareware utility available in the General Utilities library in CompuServe's IBMSYS forum. It measures SMARTDrive's "hit" rate, or the number of times the hard drive controller finds the information it is looking for in the cache.)

SMARTDrive always leaves some extended memory, because some large applications (such as Borland International's Paradox for DOS) require extended memory to run. If you're sure your applications don't require extended memory, though, you can "force" SMARTDrive to use a larger cache. It's simply a matter of specifying the cache size in the command line. For example, with 2MB of extended memory, the program sets up a 1MB cache. To have it use all 2MB as cache, edit your AUTOEXEC.BAT file as follows:

```
C:\DOS\SMARTDRV.EXE 2048
```

If you don't want to use SMARTDRV's default size, you can specify any size (up to the limit of your extended memory, of course).

You also can specify the minimum cache size by adding a second number to the command line. Why would you want to do that? Some badly behaved DOS applications (mainly old games from the mid-1980s) grab control of extended memory and can squeeze the cache, reducing its size to a point where it's no longer useful. The fol-

DOS
TIP

BEST BUFFERS

For the best performance when using SMARTDrive, set BUFFERS=3 in your CONFIG.SYS file. Extra buffers take up unneeded space in your base memory and can slow cache performance. Some applications require that BUFFERS= be set to a higher number. If that's the case, you don't have much choice but to follow your application's lead.

—S.M.

Extended Memory

(above 1 MB) in your PC

DOS Cache**Windows Cache**

1MB	1MB	none
2MB	1MB	256K
4MB	2MB	512K
6MB	2MB	1MB
> 6MB	2MB	2MB

Figure 1. Windows handles its caching differently from DOS, changing the cache size depending on its resource needs.

lowing sample command solves this problem:

```
C:\DOS\SMARTDRV.EXE 2048 512
```

It specifies a cache size of 2,048MB and indicates that the cache can never get smaller than 512K.

Tuning Options

Although adjusting the size of the cache and deciding whether or not to use write caching are the major decisions that can improve SMARTDrive performance, the program also comes with a variety of options and switches. You can use all of them from within AUTOEXEC.BAT, although some are called most often from the command line. Note: An asterisk indicates options that are available only with SMARTDrive 5.0, which comes with DOS 6.2. To determine which version is installed in your machine, type SMARTDRV /S.

OPTION: *drive letter*
drive letter+
drive letter-

FUNCTION: Specifies the letter of the drive for which you want to control caching. If you omit this option, SMARTDrive read-caches all floppy disk drives and CD-ROM drives, and both read- and write-caches hard disks and compressed disks. It won't cache RAM disks (created by RAMDISK.SYS) or network drives. If you enter the drive letters of these disk types on SMARTDrive's command line, the program will ignore them.

If you specify just a drive letter with neither a plus or minus sign, normal (read) caching is turned on and write caching is turned off for that drive. If you follow the drive letter with a plus sign, both read and write caching are turned on. A drive letter followed by a minus sign turns off all caching for that drive.

For example, the following command:

```
SMARTDRV A B C+ D-
```

turns on read caching for the A and B (floppy) drives, turns on read and write caching for the C drive, and

turns off both caching types for drive D.

OPTION: */E: size*

FUNCTION: Element size—the amount of the cache in bytes that SMARTDrive moves at one time. The default is 8192 (8K), which is the maximum size allowed. You can use only four values: 1,024, 2,048, 4,096, and 8,192. Normally, you don't need to change this. However, smaller values for the element size require less RAM

for SMARTDrive to run.

To change the element size, for example, you might issue this command:

```
SMARTDRV /E:1024
```

OPTION: */B: size*

FUNCTION: The size of the read-ahead buffer, which is the additional information SMARTDrive reads and saves to RAM after your PC requests data from a disk. The default size is 16,384 bytes (16K). The buffer size must be a multiple of the element size.

You should note that using a larger read-ahead buffer can increase SMARTDrive performance, especially if you're working with large files (a database or graphics application, for example). The downside is the larger the buffer size, the more RAM that is needed to run SMARTDrive.

OPTION: */C*

FUNCTION: If you have turned on write caching, use this option from the command line to make sure all cached information in RAM is written to disk. Typically, you would use this before turning off your PC, although it's unnecessary as long as you wait a few seconds to shut down your system. If you want to be absolutely safe, a short batch file, such as the following SHUTDOWN.BAT, can automate the process:

```
@REM CLEAR CACHE BEFORE SHUTDOWN
C:\DOS\SMARTDRV /C
CLS
ECHO System Ready for Shutdown
```

OPTION: */R*

FUNCTION: Option /R is like /C, but it restarts SMARTDrive after clearing the cache. It writes all cached data to disk, removes SMARTDrive from memory, and then starts the program again with the same options contained in your AUTOEXEC.BAT file. Since this effectively returns the program to its power-up status, you can

use /R to reset SMARTDrive after you've experimented with other options.

OPTION: */F

FUNCTION: Writes all cached data to disk after a command is completed. It is a SMARTDrive default. If you don't want to take the slim chance of losing data by using the program's full-fledged write caching, this option is one of the safest available. With /F, all cached data is written to disk whenever you exit a program or complete a DOS command.

OPTION: /L

FUNCTION: Forces SMARTDrive to load itself into low memory if the UMBs aren't available or are being used by other applications when you install DOS. (Otherwise, it loads itself into the UMBs.) Under certain circumstances, using /L improves overall system performance when double buffering is required. (For more about this, see the section titled "The Double-Buffer Question.")

OPTION: */N

FUNCTION: Writes cached data to disk when your system is idle. This enables SMARTDrive's write caching for maximum speed. Instead of waiting for data to be written to the hard disk before continuing your work, the program temporarily stores it in RAM and waits for idle processor cycles before writing (between keystrokes, for example). Depending on how much work your PC is doing, all data usually is written to disk within a few seconds.

OPTION: /Q

FUNCTION: SMARTDrive won't display its status at startup. Without option /Q, SMARTDrive displays a status screen as soon as it runs from your AUTOEXEC-

drive	Disk-Caching Status		
	read cache	write cache	buffering
A:	yes	no	no
B:	yes	no	no
C:	yes	yes	no

Figure 2. Typing SMARTDRV /S at the DOS prompt gives you a SMARTDrive status report. The right-hand column is useful in determining whether you need to activate SMARTDrive's double-buffering option.

.BAT. The screen shows the size of the cache, which drives are cached, whether read and write caching are enabled, and whether double buffering is needed.

OPTION: /S

FUNCTION: Status. Used from the command line or in AUTOEXEC.BAT. For example, the command SMARTDRV /S shows detailed SMARTDrive status information. (See the /Q and /V options.) Figure 2 shows a sample output.

OPTION: */U

FUNCTION: Turns off caching for CD-ROM drives. If you don't use this option, CD-ROM drives are cached automatically. You still can turn on or off CD-ROM drive caching by using the drive letter along with a plus or minus sign. (Write caching doesn't work with a CD-ROM drive because it's a read-only device.)

By the way, to use SMARTDrive's CD-ROM caching, the Microsoft CD-ROM Extensions file (MSCDEX.EXE) must be loaded *before* SMARTDRV.EXE in your AUTOEXEC.BAT file, which you must do by editing AUTOEXEC.BAT. DOS's Setup program always puts SMARTDRV.EXE before commands.

OPTION: */V

FUNCTION: Displays detailed status information when SMARTDrive starts. (See the /Q and /S options.) You may not use both /V (verbose) and /Q (quiet) at the same time.

OPTION: */X

FUNCTION: Disables write caching for all drives. DOS 6.2x loads this option during installation. You still can enable write caching for individual drives by using the drive letter followed by the plus sign. For example, SMARTDRV /X C+ turns off write caching for all drives, except

DOS
TIP

MULTIPLE SMART'S

If your attempt to display SMARTDrive status (by typing SMARTDRV /S) gives you an "incorrect version" error, it means an older SMARTDrive version resides somewhere in your path. Either delete the older version or rename it something like SMARTDRV.OLD.

—S.M.

drive C. If you omit this switch, SMARTDrive performs write-delayed caching on the hard disk.

The Double-Buffer Question

Although the latest, greatest versions of SMARTDRV.EXE load from your AUTOEXEC.BAT file, you might, after installing DOS, run across an additional line, like this, in your CONFIG.SYS file:

```
DEVICE=C:\DOS\SMARTDRV.EXE /DOUBLE_BUFFER
```

This activates SMARTDrive's double-buffering option.

Who needs to use double buffering? Those who run Windows in Enhanced mode (as most Windows users do) or use DOS's EMM386 memory manager, and whose PC has certain SCSI (Small Computer System Interface) host adapters, certain ESDI (Extended System Device Interface) hard drives (which are rare these days), or MCA (Micro Channel Adapter) drives found in genuine IBM PS/2 computers. The majority of today's desktop PCs use IDE hard drives, which don't require double buffering.

The reason some systems require double buffering to use SMARTDrive safely is complicated. But it has to do with the way certain hard disk controllers (SCSI host adapters or ESDI and MCA) use system memory.

If you don't run Windows or use EMM386, or if you do, but don't have one of the SCSI host adapters or drives listed above, you don't need double buffering. In that case, you could remove it from your CONFIG.SYS file if DOS has installed it there. This will save RAM space, since you can run it only in low memory.

So, how do you know when you need to use SMARTDrive's double-buffering option? Luckily, when you install DOS, the SETUP (installation) program usually can figure it out. But if you would like to experiment, add the following line to your CONFIG.SYS file—if it's not there already—and reboot your system (make sure the SMARTDRV.EXE command is in your AUTOEXEC.BAT file):

```
DEVICE=C:\DOS\SMARTDRV.EXE /DOUBLE_BUFFER
```

After your PC restarts, type SMARTDRV /S, and you'll

see a display that looks something like Figure 2.

If all entries under "buffering" are "no," you don't need the double-buffering option. If any say "yes" (and you run Windows and/or use EMM386), you do. A third possibility exists: You might see a "-" under the buffering column. This means SMARTDrive can't determine whether double buffering is needed. In this case, play it safe. Let SMARTDrive do double buffering. At most, you will lose a little bit of speed—a small price to pay for the safety of your data.

Finally, if you require double buffering and your PC seems to be running slowly, try loading SMARTDrive low by using the /L option in your AUTOEXEC.BAT file.

Gauging Success

A little tweaking and experimenting with SMARTDrive can make real differences in overall system performance. Unfortunately, the program has always lacked one essential tool: an objective way to measure its performance. Some third-party caching utilities, such as The Norton Utilities' Norton Cache, come with programs that show how little changes in setting up your cache can make large performance differences.

You *can* measure the hit rate, if not the actual speed, of SMARTDrive with the aforementioned HITS62 program. Even with HITS62, however, you'll have to use the trial-and-error approach to judge SMARTDrive's performance. Altering SMARTDrive options often produces subtle differences, and it might be difficult to tell how effective are the changes. Nevertheless, you can use subjective measurements. For example, you usually can tell if your PC "feels" faster.

A simple yet effective way to gauge changes is to watch your PC's hard disk light when you open an application, call up a file, or run a DOS command. If the light doesn't come on when you press Enter, that means data is being read from the cache instead of from the disk, and SMARTDrive is working effectively.

Even with its raft of options, increasing the size of the SMARTDrive cache or using write caching can pay the highest dividends. Of course, some variables exist. Applications use files differently, and this can affect cache speed. For instance, if you use lots of small files (short documents, for example) and a small cache (such as 512K), your system may run faster than if you use large files (like graphics images) and a maximum-size (2MB) cache.

The bottom line: SMARTDrive always improves system performance, at least a little; if you use an assortment of DOS applications, such as word processors and spreadsheets, you'll get at least a 200-percent improvement in system performance—often considerably more. ■

—S.M.

DOS
TIP

MAXIMUM SMARTDRIVE

To enable all the features of SMARTDRV 5.0 (shipped with DOS 6.2), place the command C:\DOS\SMARTDRV.EXE /N /V in your AUTOEXEC.BAT file. This turns on write caching and displays detailed information at start-up.

SCANDISK, AN MS-DOS Self-Starter

ScanDisk can be like a six-year-old who's constantly looking for your attention. Maximize your productivity by setting up ScanDisk to run unattended.

by Doug Lowe

THE CORNERSTONE OF THE MS-DOS 6.2 UPGRADE IS SCANDISK, A disk repair program designed to replace the aging CHKDSK command. ScanDisk is an overdue addition to MS-DOS, and an essential tool for users who were able to buy Microsoft's new disk compression program, DoubleSpace, before it was removed from the market. □ ScanDisk is a kind of traffic cop, policing where data is parked on your disk. ScanDisk seeks out blemishes in your hard disk's surface,

identifying "bad sectors"—those sections of the disk that, in ScanDisk's opinion, are no longer reliable. ScanDisk ropes off these areas and makes every effort to relocate any data it finds in a damaged area to an undamaged part of the disk.

ScanDisk has but one flaw, and that's its insistence on your undivided attention while it works. When ScanDisk encounters a problem with your disk, it politely asks you if you want the problem fixed. Trouble is, this degree of politeness wears on you. It's like taking

your car in for service and having the mechanic call you back for permission to change the oil; then, call back later for permission to change the spark plugs.

ScanDisk's incessant prompting for authorization is particularly irritating when you want to couple it

with some other disk utilities in a single batch file that might check your disk for errors (ScanDisk), check for viruses (MSAV), and defragment your disk (DEFRAG). When combined, these chores can take hours, especially if you're a DoubleSpace user. The last thing you

ScanDisk: ScanDisk is a diagnostic and repair tool that analyzes the reliability of your storage media—hard disks and floppies. ScanDisk has the ability to lift data from so-called "bad sectors" on your disks and move them to reliable sectors, thus reducing the chances of data loss. It is an essential companion tool to DoubleSpace, MS-DOS 6.0 and 6.2's compression utility.

Command Type: External

Available: MS-DOS 6.2

want is to come into the office the next morning—having started your batch file the evening before—and find that your computer has been sitting idle all night waiting for your permission to check the DoubleSpace host drive or permission to repair a simple FAT problem.

Good news: You can set up ScanDisk so it runs unattended, checking every aspect of your hard disk and fixing any problems it encounters. Unfortunately, it's not as easy as it should be.

If you're willing to roll up your sleeves and stuff a pocket protector in your shirt, you'll learn how to customize ScanDisk for unattended operation by looking into the cavernous abyss of ScanDisk's INI file. This may get a bit technical, but stay with it. If you've only recently bought MS-DOS 6.2, you actually have version 6.21, which does not come with DoubleSpace. ScanDisk still provides worthwhile maintenance on uncompressed drives.

Switch off Those Prompts!

The first step to taming ScanDisk is learning its command-line switches. (See "ScanDisk Syntax" for a description of these switches.) To call ScanDisk for unattended operation from a batch file, you'll want to use the /ALL, /AUTOFIX, /SURFACE, and /NOSUMMARY switches as follows:

```
SCANDISK /ALL /AUTOFIX /SURFACE /NOSUMMARY
```

This command instructs ScanDisk to check all disk drives (including DoubleSpace host drives), to fix any disk problems it detects, to perform a surface scan for each drive, and to skip the summary screen that would otherwise be displayed after each drive is checked.

If you don't want to check all your drives, substitute a drive letter for the /ALL switch. For example:

```
SCANDISK C: /AUTOFIX /SURFACE /NOSUMMARY
```

If you want to check several drives in this way, just include a SCANDISK command for each drive you want to check in your batch file. For example:

```
SCANDISK C: /AUTOFIX /SURFACE /NOSUMMARY
SCANDISK E: /AUTOFIX /SURFACE /NOSUMMARY
```

In the above sequence, drives C and E are checked.

I recommend the /ALL switch, as long as you're setting up ScanDisk to run unattended.

One other switch you should consider using is /NOSAVE. It tells ScanDisk what to do when it discovers lost clusters—portions of files that unexpectedly have become separated from the file that owns them. /NOSAVE is meaningful only if you also use the /AUTOFIX switch. If you specify both /AUTOFIX and /NOSAVE, ScanDisk deletes any lost clusters it finds. If you specify /AUTOFIX but not /NOSAVE, each chain of lost clusters is converted to a file and assigned the filename FILEXXXX.CHK (for example, FILE0001.CHK, FILE0002.CHK, and so on). You then can examine those files to determine whether they contain important information.

I discovered long ago that the lost clusters rarely contain important data, and even when they do, there's usually no way to string them back together in a useful way. So I just delete them. /NOSAVE saves you the time of looking at them, wondering what the heck they are.

Looking at the Log

When ScanDisk completes its rigorous disk tests, it normally displays a summary screen that tells you

SCANDISK SYNTAX

SCANDISK [*drive letter*] [*switches*]

The *drive letter* specifies which drive to check. If you omit it, ScanDisk checks the current drive, unless you use the /ALL switch.

/ALL	Checks all drives.
/AUTOFIX	Automatically fixes any errors that are discovered. (You can't use /AUTOFIX and /CHECKONLY together.)
/CHECKONLY	Checks for errors but does not fix any errors it discovers.
/CUSTOM	Tells ScanDisk to use the Custom section settings in the SCANDISK.INI file.
/MONO	Tells ScanDisk to use a monochrome display instead of color.
/NOSAVE	Automatically deletes lost clusters. Used with /AUTOFIX. If you use /AUTOFIX without /NOSAVE, lost clusters are saved to files in the same manner as the CHKDSK command.
/NOSUMMARY	Suppresses the summary screen displayed when ScanDisk finishes checking a drive.
/SURFACE	Forces ScanDisk to perform a surface analysis.

—D.L.

how things went. But when you use the /NOSUMMARY switch, you suppress that screen. How, then, do you determine whether any errors were detected on your disk? There are two ways: by examining the ScanDisk log file or by testing in your batch file ScanDisk's exit code, the number that signals to DOS that ScanDisk has completed its job successfully.

ScanDisk writes detailed information about the checks it performs and stores the results of those checks in a file named SCANDISK.LOG, which lives in the root directory of your C drive. You can view the contents of this file by using a MORE command, like this:

```
MORE <C:\SCANDISK.LOG
```

Or, you can print it by adding the following command to your batch file:

```
COPY C:\SCANDISK.LOG PRN
```

If you want to print or display the log file only if a disk error was detected, you can use an IF ERRORLEVEL command to test the exit code issued by ScanDisk. The sidebar "ScanDisk Exit Codes" lists all the exit codes that might be emitted by ScanDisk. If the exit code is something other than zero, it means that an error was discovered, or that ScanDisk was interrupted before it could finish. The best way to handle any non-zero exit code is to print the

SCANDISK EXIT CODES

The SCANDISK command issues an exit code that you can test in a batch file with ERRORLEVEL statements. The possible values for the exit codes are:

0	No problems detected.
1	Syntax error on command line.
2	Not enough memory or other internal error.
3	Interrupted by user.
4	Surface scan interrupted by user. No errors discovered during logical tests.
254	Problems found but corrected.
255	Problems found but not corrected.

If you're writing a batch file that tests these exit codes, you'll have to consider how each one should be handled. The simplest, and often the best, thing to do is to print or display the SCANDISK.LOG file if any exit code other than zero is issued. Here's a batch file that does just that:

```
SCANDISK /ALL /AUTOFIX /NOSAVE /NOSUMMARY
IF ERRORLEVEL 1 COPY C:\SCANDISK.LOG PRN
```

—D.L.

log so you can examine it and figure out what went wrong. To do that, add this line following the SCANDISK command in your batch file:

```
IF ERRORLEVEL 1 COPY C:\SCANDISK.LOG
```

This command prints the error log if the exit code is any value greater than zero.

Custom Settings

ScanDisk's switches allow you to set up ScanDisk for automatic operation, but there are several aspects of ScanDisk that you can't control via command-line switches. For example, you can't tell ScanDisk to check a DoubleSpace host drive using command-line switches alone, nor can you tell it to make two or more passes during the surface scan for more thorough testing. However, to control these options, you'll need to jump into the abyss and edit the SCANDISK.INI file.

SCANDISK.INI is a text file that lives in your \DOS directory. To bring it up in the DOS editor, follow these steps:

- Switch to the \DOS directory
- Make a backup copy of SCANDISK.INI by using a command such as this (don't skip this step!):

```
COPY SCANDISK.INI SCANDISK.OLD
```

- Bring the file into the MS-DOS editor:

```
EDIT SCANDISK.INI
```

- Make whatever changes you want. (My suggestions are below.)
- Save your changes by pressing Alt-F, S. Then exit by pressing Alt-F, X.

The illustration file "SCANDISK.INI File and Settings" gives a partial listing of the contents of the SCANDISK.INI file. If you're a Windows user, you'll notice right away that the file is similar in structure to a Windows INI file. Fortunately, the file itself is well commented (all of the lines that start with semi-colons are comments), so you should be able to determine the meaning of each option by reading the commentary in the file.

The SCANDISK.INI file contains two sections. The first, named Environment, has five options that affect ScanDisk's general behavior. The options themselves follow the first group of comment lines and the [ENVIRONMENT] line. To change one of the options, simply edit the value following the equal sign (=). For exam-

ple, to disable mouse support, edit the Mouse line so it looks like this:

```
Mouse          = Off      ; On, Off
```

Notice the comment at the end of each line; it tells you what the valid settings for each option are.

The options in the second, or Custom, section of SCANDISK.INI are useful when you want to make ScanDisk less interactive than it normally is, but you're not quite willing to let it run completely unattended. For example, you might want to set up ScanDisk so that it fixes lost clusters and cross-linked files, since those are common error conditions, but prompts you for other types of less common and potentially more serious errors. To do that, you must carefully examine the Custom section options and set each one to your liking.

To activate the options included in the Custom section of SCANDISK.INI, you must use the /CUSTOM switch, as in:

```
SCANDISK /ALL /CUSTOM
```

When you use /CUSTOM, you probably won't want to use /AUTOFIX, /NOSAVE, and /NOSUMMARY, because those switches override equivalent settings in the SCANDISK.INI file. You do need to specify either /ALL or a drive letter, though, because SCANDISK.INI does not contain any options that tell ScanDisk which drives to scan.

If you do use the /CUSTOM switch, you probably will want to edit the SCANDISK.INI file extensively. For starters, you'll want to change the first group of CUSTOM options as follows:

```
DriveSummary  = Off
AllSummary    = Off
Surface       = Always
CheckHost     = Always
SaveLog       = Overwrite
Undo          = Never
```

This will force ScanDisk to perform a surface scan and check the DoubleSpace host (if DoubleSpace is active), suppress any prompts and summary screens, and create a new log file (a sample log file is shown in the sidebar "The SCANDISK.LOG File").

You also may want to change the options that tell ScanDisk what to do when it detects specific problems. The default setting for all of these options is Prompt, which requires the user to tell ScanDisk what to

do. You'll want to change the settings for any problems you want ScanDisk to fix. For example, to tell ScanDisk to fix cross-linked files automatically, change the Crosslinks line as follows:

```
Crosslinks     = Fix      ; FAT-level crosslinks
```

For lost clusters, you have the option of saving them in a FILExxxx.CHK file or deleting them. Change the last option (LostClust) to Save or Delete accordingly.

Most of the other options in the Custom section represent unusual errors for which you'll want ScanDisk to prompt you. When one of these errors occurs, you can access ScanDisk's extensive on-line help for a detailed description of the problem and the corrective action ScanDisk will take.

Of course, you could change every one of the Custom options to "Fix" for completely unattended operation. But if you do that, you may as well omit the /CUSTOM switch and use the /AUTOFIX switch instead.

Conclusion

As you can see, ScanDisk is among the most flexible of MS-DOS commands. Few other commands give

THE SCANDISK.LOG FILE

Here's a typical SCANDISK.LOG file, which is created in the root directory of drive C. Notice how ScanDisk reports any disk problems that were found.

```
Log file generated at 08:51AM on Tuesday, February 15,
1994.
ScanDisk checked drive C for problems, with the following
results:
DoubleSpace volume file integrity
    ScanDisk did not find any problems.
DoubleSpace volume boot sector
    ScanDisk did not find any problems.
Directory structure
    Directory entry C:\-DOC3948.TMP had an incorrect size.
    ScanDisk successfully corrected the directory entry.
MS-DOS file allocation table
    ScanDisk did not find any problems.
MS-DOS file system
    The following files were crosslinked:
        C:\-DOC3948.TMP was first crosslinked on
        cluster 824
    ScanDisk successfully resolved all crosslinks.
    There was one lost cluster.
    ScanDisk saved the lost cluster chain as a file.
DoubleSpace file allocation table
    ScanDisk did not find any problems.
Surface scan
    ScanDisk did not perform any tests.
```

—D.L.

you the choice of completely interactive operation, completely unattended operation, or anything in between.

Microsoft's use of INI files represents an emerging trend in its latest version of MS-DOS. Provided this

useful operating system does not become forever entangled in Windows, I believe that we can look forward to greater degrees of customization for any number of MS-DOS commands through setup files or INI files. ■

SCANDISK.INI FILE AND SETTINGS. Here's a partial listing of the SCANDISK.INI file. The comments after the semicolons explain the workings of each option.

```
; SCANDISK.INI
;
; This file contains settings you can use to
; customize the ScanDisk program.
;-----
; The [ENVIRONMENT] section contains the following
; settings, which determine general aspects of
; ScanDisk's behavior:
;
; Display      Configures ScanDisk to run with a
;               particular type of display. The
;               default display type is Auto (ScanDisk
;               adjusts to the current display).
;
; Mouse        Enables or disables mouse support.
;               The default value is On.
;
; ScanTimeOut  Determines whether ScanDisk should
;               detect disk timeouts while
;               performing a surface scan. The
;               default value is Off.
;
; NumPasses    Determines how many times ScanDisk
;               should check each cluster during a
;               surface scan. The default value is 1.
;
; LabelCheck   Determines whether ScanDisk should
;               check volume labels for invalid
;               characters. The default is Off.

[ENVIRONMENT]
Display      = Auto    ; Auto, Mono, Color, Off
Mouse        = On      ; On, Off
ScanTimeOut  = Off     ; On, Off
NumPasses    = 1       ; 1 through 65,535 (anything
                        over 10 is slow)
LabelCheck   = Off     ; On, Off

;-----
; The [CUSTOM] section determines ScanDisk's
; behavior when ScanDisk is started with the
; /CUSTOM switch. You can adjust these settings to
; create a customized "version" of ScanDisk. This
; can be especially useful for running ScanDisk from
; a batch file. The [CUSTOM] settings are:
;
; DriveSummary Determines whether ScanDisk displays
;               full-screen summary information
;               after checking each drive. The
;               default is Auto (ScanDisk displays
;               the summary only if it encounters
;               errors on that drive).
;
; AllSummary   Determines whether ScanDisk
```

```
;               displays full-screen summary
;               information after checking all
;               drives. The default is Auto
;               (ScanDisk displays the summary
;               only if it encounters errors on
;               any drive).
;
; Surface      Determines whether ScanDisk will
;               perform a surface scan:
;               Never (Default) Does not
;               perform a surface scan.
;               Always Performs a surface
;               scan without prompting
;               first.
;               Prompt Prompts before performing
;               a surface scan.
;               The /SURFACE command-line switch
;               overrides this setting.
;
; CheckHost    Determines whether ScanDisk will
;               first check a host drive before
;               checking any compressed drives
;               located on that drive:
;               Never (Default) Does not check
;               the host drive.
;               Always Checks the host drive
;               without prompting first.
;               Prompt Prompts before checking
;               the host drive.
;
; SaveLog      Determines what ScanDisk does with
;               the repair log file:
;               Off (Default) Does not
;               save the repair log.
;               Append Appends the log to the
;               previous log, if any.
;               Overwrite Replaces the previous
;               log with the new log.
;
; Undo         Determines whether ScanDisk creates
;               an Undo floppy disk. The default is
;               Never (ScanDisk does not create an
;               Undo disk). The Prompt value causes
;               ScanDisk to prompt you for a disk.

[CUSTOM]
DriveSummary  = Auto    ; Auto, On, Off
AllSummary    = Auto    ; Auto, On, Off
Surface       = Never   ; Never, Always, Prompt
CheckHost     = Never   ; Never, Always, Prompt
SaveLog       = Off     ; Off, Append, Overwrite
Undo          = Prompt  ; Prompt, Never
```

End of illustration file

OUT OF SIGHT, Out of Mind?

These helpful reminder programs won't let you forget to run
your MS-DOS 6.x disk maintenance utilities.

by John Wolfskill

HAS YOUR HARD DRIVE GONE A LITTLE FLABBY? SOMEWHAT BREATH-
less after a few routine disk calls? A hard drive that's paunchy with data duplications and bloated by file fragments is a high-risk candidate for heart failure. In fact, high-performance computing begins with a well-toned drive. Until recently, you had to invest in extra software to keep your drive and its files physically fit. MS-DOS, itself, contained no substantial diagnostic or maintenance programs.

MS-DOS 6.0 and 6.2 add a number of utilities (MS-BACKUP, DEFRAG, and, in 6.2, SCANDISK) that, working together, can keep your hard drive ticking in double time. With these new tools at your disposal, your only enemy is a casual attitude toward hard disk maintenance.

And that's where MAINT62.BAT or MAINT60.BAT can help. Working with a complementary QBasic program, these programs are a constant reminder to perform disk maintenance on a schedule you set. (The programs you enter, and their names, will vary depending on whether you own MS-DOS 6.2 or MS-DOS 6.0, but for the sake of clarity, I'll refer to MAINT62.BAT and MAINT62.BAS.)

When attached to your AUTOEXEC.BAT file, MAINT62.BAT displays a menu of DOS's disk maintenance programs, lets you launch MSBACKUP, DEFRAG, SCANDISK, or CHKDSK, and, day-by-day, gently reminds you with an

"overdue" notice when it's time to perform some aspect of maintenance. (For a description of each of these programs, see the sidebar "Hard Disk Housekeeping.")

But don't think MAINT62.BAT will get in the way. Whether or not disk maintenance is overdue, MENU.BAT will return to processing your AUTOEXEC.BAT file if you don't select a menu item within 15 seconds.

Typing and Using MAINT62.BAT

If you use MS-DOS 6.2, type in the first listing, then save it as MAINT62.BAT. If you use MS-DOS 6.0, type in the second listing and label it MAINT60.BAT. Move or copy the file to the directory where you store QBASIC.EXE (usually C:\DOS), then type:

```
QBASIC MAINTxx.BAS
```


If your operating system is MS-DOS 6.2, carefully type in and save the third listing as MAINT62.BAS. (If you use MS-DOS 6.0, type in and save the fourth listing as MAINT60.BAS.)

Exit the QBasic editor to the DOS system prompt. Check to make sure the following utility programs are stored in your DOS directory: CHKDSK.EXE, SCANDISK.EXE, DEFRAG.EXE, and MSBACKUP.EXE.

Now load AUTOEXEC.BAT into your favorite ASCII text editor to make the modifications that will allow MAINT62.BAT to start automatically each time you reboot your PC. Add the following line to your AUTOEXEC.BAT:

```
CD\DOS
CALL MAINT62.BAT
CD\
```

Next, save the modified file to disk. If you choose not to add MAINT62.BAT to your AUTOEXEC.BAT file, you can start it from the system prompt in the DOS directory. MAINT62.BAS creates several data files for its own use; therefore, you must always log into the \DOS directory before you attempt to run the program.

Something to keep in mind: MAINT62.BAT serves only as a reminder to perform disk maintenance tasks. The program doesn't actually determine whether a maintenance action has been performed. One other caveat: MAINT62.BAS will reset all menu items to "overdue" on the first computing session of a new calendar year.

Although CHKDSK.EXE and CHKDSK.EXE /F appear as menu items in MAINT62.BAT, the program doesn't track CHKDSK usage. Ideally, you should run CHKDSK each time you start your computer to confirm the integrity of your directory structures and files. At times, DEFRAG.EXE also may instruct you to run CHKDSK (or SCANDISK if you have DOS 6.2) prior to starting a file defragmentation session.

How MAINT62.BAT and MAINT62.BAS Work Together

Because DOS allows batch files to make only rudimentary decisions, MAINT62.BAT enlists the help of a QBasic utility to perform some of its chores. For example, a batch file under DOS has no way to perform math. For that reason, MAINT62.BAS records the current date using QBasic's DATE\$ function, then converts the resulting month-day-year variable to a predefined

MAINT62.BAT, the maintenance batch-file listing for MS-DOS 6.2 users. It works in tandem with MAINT62.BAS.

```
:START
@ECHO OFF
QBASIC /RUN MAINT62.BAS
```

```
CLS
SET FRAGCHECK=CURRENT
SET SCANCHECK=CURRENT
SET BACKCHECK=CURRENT
```

```
IF EXIST FRAG.DUE SET FRAGCHECK=OVERDUE
IF EXIST SCAN.DUE SET SCANCHECK=OVERDUE
IF EXIST BACK.DUE SET BACKCHECK=OVERDUE
```

```
ECHO.
ECHO --- Disk Maintenance Reminder Utility ---
ECHO * DISK DEFRAGMENTATION IS %FRAGCHECK% *
ECHO * DISK ANALYSIS AND REPAIR CHECK IS
    %SCANCHECK% *
ECHO * DISK BACKUP IS %BACKCHECK% *
ECHO -----
ECHO.
SET FRAGCHECK=
SET SCANCHECK=
SET BACKCHECK=
```

```
ECHO A   CHKDSK   - RUN CHKDSK
ECHO B   CHKDSK   - RUN CHKDSK /F
ECHO C   DEFRAG   - Defragment hard drive(s)
ECHO D   SCANDISK - Analyze/repair files & disk
                    surface
```

```
ECHO E   MSBACKUP - Backup drive(s) data
ECHO F   EXIT     - to DOS prompt
ECHO.
CHOICE /T:F,15 /C:ABCDEF Please choose a
maintenance option :
```

```
IF ERRORLEVEL 6 GOTO ENDIT
IF ERRORLEVEL 5 GOTO BACKUP
IF ERRORLEVEL 4 GOTO SCAN
IF ERRORLEVEL 3 GOTO DEFRAG
IF ERRORLEVEL 2 GOTO CHKDSKF
IF ERRORLEVEL 1 GOTO CHKDSK
```

```
:CHKDSK
CLS
CHKDSK
PAUSE
GOTO START
```

```
:CHKDSKF
CLS
CHKDSK /F
PAUSE
GOTO START
```

```
:DEFRAG
CLS
DEFRAG
IF EXIST FRAG.DUE SET FRAGCHECK="1"
GOTO START
```

```
:SCAN
CLS
SCANDISK
IF EXIST SCAN.DUE SET SCANCHECK="1"
GOTO START
```

```
:BACKUP
CLS
MSBACKUP
IF EXIST BACK.DUE SET BACKCHECK="1"
GOTO START
```

```
:ENDIT
CLS
```

End

MAINT60.BAT, the maintenance batch-file listing for MS-DOS 6.0 users, works with **MAINT60.BAS**.

```
:START
@ECHO OFF
QBASIC /RUN MAINT60.BAS

CLS
SET FRAGCHECK=CURRENT
SET BACKCHECK=CURRENT

IF EXIST FRAG.DUE SET FRAGCHECK=OVERDUE
IF EXIST BACK.DUE SET BACKCHECK=OVERDUE

ECHO.
ECHO --- Disk Maintenance Reminder Utility ---
ECHO * DISK DEFRAGMENTATION IS %FRAGCHECK% *
ECHO * DISK BACKUP IS %BACKCHECK% *
ECHO -----
ECHO.
SET FRAGCHECK=
SET BACKCHECK=

ECHO A   CHKDSK   - RUN CHKDSK
ECHO B   CHKDSK   - RUN CHKDSK /F
ECHO C   DEFRAG    - Defragment hard drive(s)
ECHO D   MSBACKUP  - Backup drive(s) data
ECHO E   EXIT      - to DOS prompt
ECHO.
CHOICE /T:E,15 /C:ABCDE Please choose a
      maintenance option :
```

```
IF ERRORLEVEL 5 GOTO ENDIT
IF ERRORLEVEL 4 GOTO BACKUP
IF ERRORLEVEL 3 GOTO DEFRAG
IF ERRORLEVEL 2 GOTO CHKDSKF
IF ERRORLEVEL 1 GOTO CHKDSK

:CHKDSK
CLS
CHKDSK
PAUSE
GOTO START

:CHKDSKF
CLS
CHKDSK /F
PAUSE
GOTO START

:DEFRAG
CLS
DEFRAG
IF EXIST FRAG.DUE SET FRAGCHECK="1"
GOTO START

:BACKUP
CLS
MSBACKUP
IF EXIST BACK.DUE SET BACKCHECK="1"
GOTO START

:ENDIT
CLS
```

End

date; that is, the numerical day of the year between 1 and 365.

Next, the program creates a small data file named **MAINT.DAT**, where it stores the predefined date for each maintenance item. The variables **SCAN**, **FRAG**, and **BACK** represent the last date on which the corre-

sponding maintenance task was performed.

The frequency for performing each maintenance check (daily, weekly, or monthly) is based on the values of **SCANPERIOD**, **FRAGPERIOD**, and **BACKPERIOD**. The value you assign to each variable represents the number of days between subsequent disk maintenance checks.

For example, to change **MSBACKUP** from a 30- to a 10-day frequency, modify the **BACKPERIOD** variable to read:

```
BACKPERIOD = 10
```

MAINT62.BAS calculates a maintenance overdue date for each menu item by adding the value of the **xxxxPERIOD** (**BACKPERIOD**, for example) variable to the **NEWJULIAN** variable, which represents the current day of the year. Then, the program compares the result with **SCAN**, **FRAG**, or **BACK**, all of which have been stored in **MAINT.DAT** from a previous session. If the result of a comparison is less than or equal to zero, the maintenance item is overdue. In this case, **MAINT62.BAS** cre-

**DOS
TIP**

SHORTEN THE PATH, INCREASE SPEED

You can include a program's full **PATH** as part of its **DOSKEY** macro. That means you probably can remove the program's directory from your **PATH**, and the shorter your **PATH**, the faster your machine will run. This can be a major benefit on a system with several logical drives on a large hard disk, where **PATH** statements tend to fill the 127-character command line quickly.

If you're running out of space in your **PATH** statement, try deleting a few applications directories from the **PATH** and installing **DOSKEY** macros pointing to them instead. For example, rather than including your **D:\WP\GMK** style checker in the path, make a **DOSKEY** macro pointing to it, such as:

```
DOSKEY GMK=D:\WP\GMK\G4 $*
```

It won't take much buffer space, it will execute faster than it would if DOS had to walk a long **PATH** to find it, and your **PATH** statement gains room for more entries.

—Craig Menefee & Nick Anis
Harnessing DOS 6.0

The QBasic listing MAINT62.BAS provides certain functions required by its accompanying batch file.

```

DEFINT A-Z
ON ERROR GOTO TESTERROR
CLS : DIM DAY(12)

SCANPERIOD = 30 ' Desired days between disk scan
               tests (range 1 to 365)
FRAGPERIOD = 30 ' Desired days between disk
               defragmentation (range 1 to 365)
BACKPERIOD = 7  ' Desired between disk data
               backups (range 1 to 365)
DUMMY$ = " "    ' Filler variable for .DUE
               files

'Data for number of days in each month of the year
DAY(1) = 31

IF CURRENTYEAR MOD 4 THEN ' This line accounts
    for leap years
    DAY(2) = 29
ELSE
    DAY(2) = 28
END IF

DAY(3) = 31: DAY(4) = 30: DAY(5) = 31: DAY(6) =
    30: DAY(7) = 31
DAY(8) = 31: DAY(9) = 30: DAY(10) = 31: DAY(11) =
    30: DAY(12) = 31
'Calculate the current month, day and year
CURRENTMONTH = VAL(LEFT$(DATE$, 2))
CURRENTDAY = VAL(MID$(DATE$, 4, 2))
CURRENTYEAR = VAL(RIGHT$(DATE$, 2))
'Convert the mm/dd/yy result to a Julian date
FOR JJ = 1 TO CURRENTMONTH - 1
    NEWJULIAN = NEWJULIAN + DAY(JJ)
NEXT

NEWJULIAN = NEWJULIAN + CURRENTDAY
'Open maintenance data file
'The file MAINT.DAT maintains the Julian dates that
'tell MAINT.BAT when the action (DEFRAG, SCANDISK,
'BACKUP) was last accessed. The stored values are
'compared with the current Julian date.

100 OPEN "MAINT.DAT" FOR INPUT AS #1
    INPUT #1, SCAN
    INPUT #1, FRAG
    INPUT #1, BACK
    INPUT #1, OLDYEAR
    CLOSE #1

200 IF CURRENTYEAR <> OLDYEAR THEN ERROR 99
'check for year rollover

'The lines below change the "overdue" status to
"current" when the corresponding program is
'accessed before its due date. They also then
'establish a new "due" date for the action.

300 IF ENVIRON$("SCANCHECK") > "" THEN
    KILL "SCAN.DUE": SCAN = NEWJULIAN
    IF ENVIRON$("FRAGCHECK") > "" THEN
        KILL "FRAG.DUE": FRAG = NEWJULIAN
    IF ENVIRON$("BACKCHECK") > "" THEN
        KILL "BACK.DUE": BACK = NEWJULIAN

```

'The lines below change the "current" status to "overdue" if the corresponding program is NOT 'accessed before its due date.

```

500 IF NEWJULIAN >= SCAN + SCANPERIOD THEN
    OPEN "SCAN.DUE" FOR OUTPUT AS #1
    PRINT #1, DUMMY$
    CLOSE #1
END IF

```

```

IF NEWJULIAN >= FRAG + FRAGPERIOD THEN
    OPEN "FRAG.DUE" FOR OUTPUT AS #1
    PRINT #1, DUMMY$
    CLOSE #1
END IF

```

```

IF NEWJULIAN >= BACK + BACKPERIOD THEN
    OPEN "BACK.DUE" FOR OUTPUT AS #1
    PRINT #1, DUMMY$
    CLOSE #1
END IF

```

```

'Update maintenance data file and return to
'MAINT.BAT
OPEN "MAINT.DAT" FOR OUTPUT AS #1
PRINT #1, SCAN
PRINT #1, FRAG
PRINT #1, BACK
PRINT #1, CURRENTYEAR
CLOSE #1

```

SYSTEM

TESTERROR:

'The following lines establish the initial data for 'MAINT.DAT. In addition, they reset default values 'at the beginning of each calendar year.

```

IF ERR = 53 AND ERL = 100 OR ERR = 99 THEN
    OPEN "MAINT.DAT" FOR OUTPUT AS #1
    PRINT #1, NEWJULIAN
    PRINT #1, NEWJULIAN
    PRINT #1, NEWJULIAN
    PRINT #1, CURRENTYEAR
    SCAN = NEWJULIAN
    FRAG = NEWJULIAN
    BACK = NEWJULIAN
    CLOSE #1

```

```

OPEN "BACK.DUE" FOR OUTPUT AS #1:PRINT #1,
DUMMY$:CLOSE #1
OPEN "SCAN.DUE" FOR OUTPUT AS #1:PRINT #1,
DUMMY$:CLOSE #1
OPEN "FRAG.DUE" FOR OUTPUT AS #1:PRINT #1,
DUMMY$:CLOSE #1

```

```

IF ERR = 99 THEN RESUME 300 ELSE RESUME 200
END IF

```

RESUME NEXT

End

The QBasic listing MAINT60.BAS is the helper program for MAINT60.BAT.

```
DEFINT A-Z
ON ERROR GOTO TESTERROR
CLS : DIM DAY(12)

FRAGPERIOD = 30 ' Desired days between disk
                 defragmentation (range 1 to 365)
BACKPERIOD = 7  ' Desired between disk data
                 backups (range 1 to 365)
DUMMY$ = " "    ' Filler variable for .DUE files

'Data for number of days in each month of the year
DAY(1) = 31

IF CURRENTYEAR MOD 4 THEN ' This line accounts
    for leap years
    DAY(2) = 29
ELSE
    DAY(2) = 28
END IF

DAY(3) = 31: DAY(4) = 30: DAY(5) = 31: DAY(6) =
    30: DAY(7) = 31
DAY(8) = 31: DAY(9) = 30: DAY(10) = 31: DAY(11) =
    30: DAY(12) = 31
'Calculate the current month, day and year
CURRENTMONTH = VAL(LEFT$(DATE$, 2))
CURRENTDAY = VAL(MID$(DATE$, 4, 2))
CURRENTYEAR = VAL(RIGHT$(DATE$, 2))
'Convert the mm/dd/yy result to a Julian date
FOR JJ = 1 TO CURRENTMONTH - 1
    NEWJULIAN = NEWJULIAN + DAY(JJ)
NEXT

NEWJULIAN = NEWJULIAN + CURRENTDAY
'Open maintenance data file
'The file MAINT.DAT maintains the Julian dates that
'tell MAINT.BAT when the action (DEFRAG OR BACKUP)
'was last accessed. The stored values are compared
'with the current Julian date.

100 OPEN "MAINT.DAT" FOR INPUT AS #1
    INPUT #1, FRAG
    INPUT #1, BACK
    INPUT #1, OLDYEAR
    CLOSE #1

200 IF CURRENTYEAR <> OLDYEAR THEN ERROR 99
'check for year rollover

'The lines below change the "overdue" status to
'"current" when the corresponding program is
'accessed before its due date. They also then
'establish a new "due" date for the action.

300 IF ENVIRON$("FRAGCHECK") > "" THEN KILL
```

```
"FRAG.DUE": FRAG = NEWJULIAN
IF ENVIRON$("BACKCHECK") > "" THEN KILL
"BACK.DUE": BACK = NEWJULIAN
```

'The lines below change the "current" status to "overdue" if the corresponding program is NOT accessed before its due date.

```
500 IF NEWJULIAN >= FRAG + FRAGPERIOD THEN
    OPEN "FRAG.DUE" FOR OUTPUT AS #1
    PRINT #1, DUMMY$
    CLOSE #1
END IF
```

```
IF NEWJULIAN >= BACK + BACKPERIOD THEN
    OPEN "BACK.DUE" FOR OUTPUT AS #1
    PRINT #1, DUMMY$
    CLOSE #1
END IF
```

```
'Update maintenance data file and return to
'MAINT.BAT
OPEN "MAINT.DAT" FOR OUTPUT AS #1
PRINT #1, FRAG
PRINT #1, BACK
PRINT #1, CURRENTYEAR
CLOSE #1
```

SYSTEM

TESTERROR:

'The following lines establish the initial data for 'MAINT.DAT. In addition, they reset default values 'at the beginning of each calendar year.

```
IF ERR = 53 AND ERL = 100 OR ERR = 99 THEN
    OPEN "MAINT.DAT" FOR OUTPUT AS #1
    PRINT #1, NEWJULIAN
    PRINT #1, NEWJULIAN
    PRINT #1, CURRENTYEAR
    FRAG = NEWJULIAN
    BACK = NEWJULIAN
    CLOSE #1
```

```
OPEN "BACK.DUE" FOR OUTPUT AS #1: PRINT #1,
    DUMMY$: CLOSE #1
OPEN "FRAG.DUE" FOR OUTPUT AS #1: PRINT #1,
    DUMMY$: CLOSE #1
```

```
IF ERR = 99 THEN RESUME 300 ELSE RESUME 200
END IF
```

RESUME NEXT

End

ates a small data file, named BACK.DUE, which it deposits on your hard disk for MAINT62.BAT to find. The process is the same for FRAG and SCAN.

QBasic then hands the baton back to DOS, where MAINT62.BAT continues with its work. Based upon the presence of a BACK.DUE, FRAG.DUE, or SCAN.DUE file, MAINT62.BAT uses the DOS SET command to create an

environment variable (here, for BACK.DUE):

```
SET BACKCHECK = "current"
```

if a BACK.DUE file was found, or:

```
SET BACKCHECK = "overdue"
```


if a BACK.DUE file was not found.

MAINT62.BAT uses the environment variable to display the word "current" or the word "overdue" for each menu item. When the program informs you that a maintenance check is overdue, it's a simple reminder to run the corresponding disk maintenance utility.

After you run the disk maintenance utility, MAINT62.BAT uses the DOS SET command to create a DOS environment variable of 1. For example, SET BACKCHECK = "1".

When MAINT62.BAS runs and finds that "1" is the value of the ENVIRON\$ variable, it resets the corresponding date variable, BACK, to the current Julian date, then erases the corresponding BACK.DUE file from your disk.

Sharing Information between MAINT62.BAT and MAINT62.BAS

It's easy to call a QBasic program from a batch file, but I had to find a way for the QBasic program to communicate with my batch file—not so easily done.

One communications path shared by both DOS and QBasic is the DOS environment table. Environment variables can be used to control the behavior of batch files and QBasic programs. During operation, MAINT62.BAT uses this technique to establish communication with MAINT62.BAS.

Because the environment table is critical to MAINT62.BAT's performance, you might want to raise its ceiling. DOS 6.x allocates 256 bytes of memory for the environment table, which includes space for your PATH statement, a prompt defined in your AUTOEXEC.BAT, as well as any other instructions you've given your system via the DOS SET command.

If you're going to use this environment space for communicating with QBasic, it may be necessary to expand its size to avoid DOS's "out of environment space" error message each time you run MAINT62.BAT.

Fortunately, it's easy to expand the DOS environment. Load your computer's CONFIG.SYS file into your word processor or text editor and add the following line to the file:

```
DOSHELL=C:\DOS\COMMAND.COM C:\DOS
/p /e:512
```

Then, save the file in ASCII format and reboot.

As indicated, the /e: switch expands the size of the DOS environment table to 512 bytes, but you can reserve more or less space depending on the length of your existing PATH and other SET variables defined in your AUTOEXEC.BAT file. ■

HARD DISK HOUSEKEEPING

Here's a brief description of the utilities MAINT62.BAT runs. All but MSBACKUP are also covered elsewhere in this issue.

CHKDSK.EXE: Although your hard drive is a highly reliable data storage device, unforeseen problems, such as power surges, program bugs, and operator errors, can wreak havoc upon your programs and data. The result can be files with lost data clusters, invalid subdirectory entries, and cross-linked File Allocation Table (FAT) entries. CHKDSK.EXE allows you to check the integrity of your disk's file and directory structure, and correct these problems before they create serious difficulties that can cause your hard disk to crash. You should run CHKDSK.EXE daily to look for disk file problems, and, if the only errors are lost clusters, follow up by using CHKDSK.EXE /F to repair the damage the utility finds.

DEFRAG.EXE: File fragmentation occurs when DOS uses disk data clusters made available by files you've erased from your disk in order to store newer files. DEFRAG.EXE defragments files by reorganizing information so that each file is stored in consecutive data clusters. The benefit of defragmentation is that your hard drive can access an entire file quickly, without having to move its read/write head repeatedly over widely scattered areas of the disk's data platter. Periodically, the files on even the best maintained hard drive will become fragmented, which can increase data access time significantly and lower your computer's overall performance. It's a good idea to run DEFRAG.EXE at least once a month.

MSBACKUP.EXE: A reliable disk backup utility is the first line of defense against disk disasters. MSBACKUP.EXE is a limited version of Symantec's The Norton Backup program. This utility allows you to archive the files on your hard drive by saving them to a series of floppy disks. In order to place a maximum amount of data onto each backup disk, MSBACKUP lets you use the same data compression engine built into the DOS 6.0 and 6.2 DoubleSpace utility. MSBACKUP lets you back up and restore selected files or directories and supports a full range of backup options.

SCANDISK.EXE: SCANDISK.EXE (available only with DOS 6.2) is a disk analysis and repair tool that thoroughly checks your drive for errors and corrects any problems it finds. Like CHKDSK.EXE, this utility can detect and repair files with lost or damaged data clusters. It also can perform a sophisticated surface scan of both uncompressed and DoubleSpace volume files to test the magnetic media for marginal areas that may be prone to fail in the future. SCANDISK automatically performs this surface scan after it checks your hard drive for cluster damage. When scanning an uncompressed drive, SCANDISK confirms that your data can be written and read from the drive reliably. During a surface scan of a DoubleSpace drive, the utility confirms that your data can be decompressed. You should run SCANDISK at least once a month on all drives. It also may be prudent to run SCANDISK immediately after you've used CHKDSK.EXE to repair file damage.

—J.W.

USING DOSKEY

Command Macros

Plug into DOSKEY, and you quickly can build DOS macros that automate your commands, eliminate drudgery, and add power to every keystroke.

by Craig Menefee and Nick Anis

THE DIFFERENCE BETWEEN ENTERING COMMANDS AT THE DOS PROMPT and entering a macro of your own design is like the difference between using a hand tool and a power tool. For all but the smallest jobs, a power tool will save both time and effort, make jobs easier, and enlarge the scope of what you can accomplish. □ DOS's command macro program, DOSKEY, is a definite power tool. The first time you enter a macro such as C:\>BACKUP and watch that simple command "nickname" trans-

form into a complex and precise DOS command, such as:

```
C:\>C:\DOS\BACKUP \*.* A: /S /M /A /F:12
```

you will know why so many people depend on DOSKEY.

The DOSKEY program's command macros are stored in memory, so they are fast and always accessible. When you type a macro's name, DOS executes the assigned keystrokes automatically, as if you had typed them yourself. The backslashes, virgules, and other symbols appear on the screen instantly, and DOS executes the macro with no further typing on your part.

In this article, we'll explain how to put together macros that automate your commands, prevent typos, reduce data loss, eliminate drudgery, and tame more dangerous DOS commands. We'll also show how to

create and use some general-purpose macros that DOS users often consider indispensable.

Tuning DOSKEY for Macros

DOSKEY is a TSR (Terminate-and-Stay-Resident) utility program that lets you recall, edit, and execute previously entered DOS commands, as well as build command macros. It runs rings around the older F-key editing system, and it's a memory miser (taking only 3K of RAM) that works fine in upper memory if your machine has it.

Your DOS manual explains the basic steps for installing DOSKEY in conventional or high memory. But for use with macros, it's a good idea to fine-tune the installation command. Most important, the default DOSKEY buffer size of 512 bytes is really not enough if

DOS 6.0 POWER MACROS

The macros listed here are on the disk included with our book, *Harnessing DOS 6.0: Batch File and Command Macro Power*. To use them, first enter the command:

```
DOSKEY /bufsize=2048
```

from the DOS prompt, or include it in your AUTOEXEC.BAT file. The commands you should assign to each macro are shown directly below its description.

You'll note that the macro names are fairly long so that they can be fully descriptive. Once you settle on some favorites, you'll save keystrokes by renaming them to something shorter. For example, FORMFEED could become FF. For faster execution, you can add paths to the program names. The tradeoff is that this takes additional memory.

—C.M. & N.A.

ADD—like APP (below), but with \$1 and \$2 reversed. Use the version that seems most natural, so you don't get the files in reverse order accidentally. ADD appends the second named file to the first named file.

```
ADD=TYPE $2 $G$G $1
```

APP—(APPend) adds one file to the end of another, using \$G\$G redirection. Note: This macro works only on "pure" ASCII files; that is, files that contain no binary code, formatting, or embedded ASCII-26 characters. APP appends the first named file to the second named file.

```
APP=TYPE $1 $G$G $2
```

CANCEL—automatically removes a specified macro from memory. (Use this to gain space or cancel a macro for the rest of a session.)

```
CANCEL=DOSKEY $1=
```

CC—provides quick change to C:\ and shows use of the pipe symbol to speed macro execution. Note: Don't use \$B pipe to

chain a command that produces output that might confuse the next command in the chain.

```
CC=C: $B CD \ $B DIR /W
```

COPC—copies a file somewhere, then compares the copy with the original. The piped ECHO N causes COMP.EXE to exit when finished.

```
COPC=COPY $1 $2 /V $T ECHO N $B COMP $1 $2
```

DIAL—lets you dial a phone number from the command line. As shown, it assumes you have a 2400-baud modem on serial port 1 (COM1). If your modem is a different speed, change the 2400 to, for example, 1200 or 9600. If your modem is on COM1, also change that entry in the macro. Usage is DIAL *number*, and when you hear the modem start to ring, pick up your receiver and press any key to take the modem back off-line.

```
DIAL=MODE COM1 2400 $TECHO ATDT; $1  
$GCOM1 $TPAUSE $TECHO ATH $GCOM1
```

DIRS—lists subdirectories of current directory, sorted, with no filenames.

```
DIRS=DIR $1 /AD /O
```

FINDALL—finds all files matching a file specification, across a whole drive.

```
FINDALL=DIR \ $1 /S/B
```

FINDPART—looks for any matching string in a verbose CHKDSK report on the current drive (but not on some other drive)—good for finding partial filenames.

```
FINDPART=CHKDSK /V $B FIND "$*" /I
```

FORMATTING: Following are three safe disk-formatting macros that assume you have renamed FORMAT.EXE to FMT.EXE (see the ZAP macros below for a faster reformat):

FMT.B.NB—unconditional format of drive B. (Note: requires you

you intend to use macros as well as the history buffer. Therefore, you should increase the default buffer size from 512 bytes to 2K. The command syntax for a 2K buffer size would be:

```
C:\>DOSKEY /BUFSIZE=2048
```

You can use round numbers. A 1K buffer will hold 10 to 20 macros plus a command history of 15 to 25 commands. A 2K buffer will hold twice that many. The actual numbers will depend upon the complexity of your macros and command lines. It is good practice to install only the macros you use regularly. That way you can save most of your RAM for your application programs.

(Note: On machines that support high memory, DOSKEY will load macros nicely. If high memory isn't

available, expect DOSKEY to take 2K to 6K of active RAM, assuming a 2K buffer.)

Creating Command Macros

Macros are simple to create. Just specify a macro name and tell DOSKEY to assign a command or series of commands to it. The formal syntax is:

```
[drive:][\path\]DOSKEY [macro-name]=[commands]]
```

where *macro-name*=[*commands*] assigns one or more [*commands*] to the command "nickname" *macro-name*. Special DOSKEY symbols can increase a macro's power and flexibility, but the basic use is simple.

DOSKEY macros have some limitations. Here are a few rules for creating them:

to create a special input file. Open an ASCII text editor, press Enter twice, type N, and save the file as `FORMAT.INP`.)

```
FMT.B=FMT B: /U $L C:\DOS\FORMAT.INP
```

FORMAT—limits formatting to 1.2MB disks in drive A. You can change “1.2” to the appropriate size, as needed.

```
FORMAT=FMT A: /F:1.2
```

REFORMAT—does /Q (quick) reformat of any size disk in drive A, but saves UNFORMAT information in case of error. Can change to REFORMAT.A.

```
REFORMAT=FMT A: /Q
```

HANGUP—hangs up phone. Assumes modem is on COM1 at 2400 baud; change as appropriate. This is handy when a program leaves the modem off-hook.

```
HANGUP=MODE COM1 2400 $T ECHO ATH $GCOM1
```

HIDE—sets (turns on) the HIDDEN attribute to keep file(s) from showing during a normal DIR command; UNHIDE resets (turns off) the HIDDEN attribute.

```
HIDE=ATTRIB +H $*
UNHIDE=ATTRIB -H $*
```

SCOPY—(SafeCOPY) uses REPLACE to copy only files that do not already exist on the target, or (with prompt) files that are newer.

```
SCOPY=REPLACE $1 $2 /A $T REPLACE $1 $2 /P/U
```

SEEMAC—lets you see the value of a specified DOSKEY macro.

```
SEEMAC=DOSKEY /M $B FIND "$1=" /I
```

SEEMACS—lets you review all loaded macros, a page at a time.

```
SEEMACS=DOSKEY /M $B MORE
```

SEESORT—sorts a file, then displays it a screen at a time —

good for viewing phone directories, for example.

```
SEESORT=SORT $L $1 $B MORE
```

SHOW—redirects a file through the MORE filter to view it one screen at a time.

```
SHOW=MORE $L $1
```

SUB—(SUBSTitute) substitutes the specified drive letter for the current directory and shows the results.

```
SUB=SUBST $1: . $T SUBST
```

SUBSOUT—cancels all SUBSTitutions. Notice how turning off ECHO reduces clutter; try it with ECHO on! NOTE: Remove letters for logical drives on your system to avoid error messages, since they'll always pass the “IF EXIST \NUL” test.

```
SUBSOUT=ECHO OFF$TFOR %V IN (D E F G H I J K L M N O
P) DO IF EXIST %V:\NUL SUBST %V: /D$TECHO ON
```

TOUCH—safely places today's date in a file, using the COPY append function.

```
TOUCH=COPY /B $1+,, $TDIR $1
```

UPDATE—copies only files that are newer than the target files.

```
UPDATE=REPLACE *.* $1 /U
```

ZAP—does /Q (quick) /U (unconditional) format. Very fast, but dangerous, because it does not save UNFORMAT information.

```
ZAP=FMT $1 /Q/U
```

ZAP2—an extremely fast version of ZAP that takes redirected input from `FORMAT.INP`. It's automatic, but very dangerous.

```
ZAP2=FMT $1 /Q/U $L C:\DOS\FORMAT.INP
```

ZAP.A and **ZAP.B**—safer versions of ZAP2 that limit action to drive A or B.

```
ZAP.A=FMT A: /Q/U $L C:\DOS\FORMAT.INP
ZAP.B=FMT B: /Q/U $L C:\DOS\FORMAT.INP
```

- Macros may be no longer than 127 characters, counting the `DOSKEY macro-name=` part of the command. That's the limit DOS places on any command line; however, this normally will not be a problem. If you use an editor or word processor to prepare your macros, you accidentally might create one that spills over the 127-character limit. In that case, DOSKEY will truncate the macro (ignore the extra characters).

- Macro names can be any length up to the 127-character command limit; the 8-character DOS command-name limit does not apply to macros. You can be as descriptive as you want when naming your macros, but short and easy-to-remember names save keystrokes and generally are best.

- Macro names are less sensitive than filenames and can include any typed character except the space.

Names containing odd characters, like `MAC.NAME` or `MAC/NAME`, are perfectly legal. Don't include any spaces within the name itself; the name `MAC NAME` would be rejected with the message “Invalid macro definition.”

Beyond the Basics

In complex and powerful macros, the dollar sign (\$) becomes very important. The \$ symbol tells DOSKEY that the next character will be an operator of some kind, such as a user-specified command parameter or a redirection or piping symbol. If you really mean to use a literal dollar sign (for example, as part of a `CD \ $MONEY` command), type two \$ symbols (for example, `CD \ $$MONEY`) in the macro definition, instead of one. Symbols like \$ are fine as part of a macro's name.

Macros can use any legal DOS command, but two,

ECHO OFF and CTTY NUL, require special care. Both reduce screen clutter, but require an additional cancelling command (ECHO ON, CTTY CON, respectively) within the same macro to restore normal operation. That makes them a drain on buffer space. CTTY is especially risky and can hang the computer if used carelessly. It is not recommended for casual use.

Macros also can run batch programs, but batch programs cannot run macros directly. A fairly simple macro can use a complex batch file as part of its definition, which can extend a macro's power considerably. When a batch program is finished, the macro that called it goes on to the next command in the macro.

Environment variables are values placed in a reserved part of memory for use by batch files and other executable programs. At the command line, a user may set environment variables but cannot use them directly in commands. Command macros work the same way; you can have a macro set an environment variable for use by a batch file, but macros cannot use environment variables directly. The command SET WP=C:\WP51 would set the expected value in the environment, but the command CD %WP% would not work correctly from the command line or from within a macro.

Finally, although most third-party command macro programs allow "nesting" macros, the current DOSKEY version does not. (*Nesting* is the term for a setup where one macro uses another macro as one of its commands.)

Using Macros Effectively

The sidebar "DOS 6.0 Power Macros" provides more than two dozen macros that you can enter by following the simple rules we've outlined. When you're done, you can see which macros you've stored in memory by typing:

```
C:\>DOSKEY /MACROS
```

The command's shortened version, DOSKEY /M, works just as well. Either form of the command will cause a list of all macros currently in memory to scroll down the screen. If you have more macros in the buffer than can be shown on a single screen, you can pipe the command through the MORE filter:

```
C:\>DOSKEY /MACROS | MORE
```

The output will pause and wait for you to hit a key each time the screen fills with new information. Save the current macro definitions using output redirection:

```
C:\>DOSKEY /MACROS > filename
```

Again, the shortened /M switch would work the same way. You can view them later using a file listing or editing program, or simply use TYPE *filename*

to have DOS show them on screen.

Sometimes a macro will have the same name as another DOS or application program command. When this is so, DOS will execute the macro instead of the standard command. To use the standard command instead, insert a space at the head of the command line when you type it. For example, you might have decided to make the DIR command display files in a wide format by defining a macro: DIR=DIR /W. Now the command C:\>DIR will always produce a wide listing. To get a normal, single-column DOS listing of programs instead, insert a space between the prompt and the command C:\> DIR.

Clearing and Cancelling Macros

DOSKEY provides two ways to delete or cancel all current macros: by reinstalling the buffer or by clearing it.

To reinstall the DOSKEY macro, enter the command:

```
C:\>DOSKEY /REINSTALL /BUFSIZE=2048
```

This will install an empty buffer of the specified size and deactivate the old buffer. However, memory taken by the original copy of DOSKEY is not returned to the system, but rather is simply withdrawn from use.

Often, users will create different sets of macros to serve different purposes, and need to change macro sets from time to time during a session. To avoid using the /REINSTALL switch, press Alt-F10, the macro-clearing "hot key." Like Alt-F7, which clears out the DOSKEY command history leaving it intact but empty, Alt-F10 leaves the current buffer clear to accept new macros without taking up extra memory.

Advanced Macro Techniques

Simple macros that do only one thing can be useful tools, acting as a sort of shorthand for commands that take a lot of parameters. And, like the format macros in the aforementioned sidebar, they also can defang the riskier DOS commands. However, to be a true enhancement, DOSKEY must handle the full range of command-line options, including user-specified command parameters. A single macro should be able to carry out a series of commands and also should handle input and output piping and redirection.

Parameters are the variable arguments that tell a command what to do. For example, in the following command line, *.DOC is the parameter: C:\>DIR *.DOC. The *.DOC parameter line tells DIR to list only those files that have .DOC as their extension. Although you could easily make that command a macro:

```
C:\>DOSKEY DIRDOC=DIR *.DOC
```

it would be limited to DOC files. Such a macro would be a waste of memory unless you have a regular need

SUMMARY OF DOSKEY USAGE

DOSKEY's different functions are separated within the table to make the information easy to find at a glance. You may wish to copy the table and keep it as a handy reference.

Not all the information here shows up in the DOSKEY help screen. In particular, note the special codes or symbols that begin with dollar signs (\$). Some of these correspond to the redirection and piping symbols. Others are place markers for command arguments supplied by the user when typing in the command.

Syntax: DOSKEY [/REINSTALL] [/BUFSIZE=*size*]
 [/INSERT or /OVERSTRIKE]
or: DOSKEY [/MACROS] [/HISTORY]
 [*Macro_name=command_sequence*]

The following keys are available at the DOS command line:

Up Arrow and Down Arrow	Make previously typed commands appear at the prompt for editing.
<ENTER>	Executes the current command.
<ESC>	Erases the current command without affecting the buffer.
F8	Recalls a command line that starts with the character or characters currently shown. To use this handy feature, type the first part of a command, and then press F8 to recall the rest.
PgDn	Recalls the most recent command line.
PgUp	Recalls the oldest command still in the buffer.
←and→	Moves the cursor one character to the left or right.
Ctrl- ←	Moves cursor to the start of the word to the left.
Ctrl- →	Moves cursor to the start of the word to the right.
Home	Moves the cursor to the beginning of the displayed command.
End	Moves the cursor to the end of the displayed command.
Insert	Toggles between insert and overwrite mode. Use the /INSERT option to start in insert mode (overwrite is default).
F7	Displays all command lines, showing their line numbers.
Alt-F7	Deletes all commands from history buffer.
F9#	Displays command-line # for editing. Press Enter to execute.
Alt-F10	Deletes all macros from buffer (use prior to loading new set).
F1...F6	Standard "template" editing keys; same as earlier DOS versions.

DOSKEY uses the following options:

/REINSTALL	Installs a new copy of DOSKEY after the program is first loaded. Use this to run a secondary DOS command-line session in Windows. Use MARK and RELEASE, instead, to change buffer size. Use Alt-F10 to clear current macros.
/BUFSIZE=#	Sets # as the size, in bytes, of the command and macro buffer. The default size is 512 bytes. Smallest is 256.
/M or /MACROS	Displays all macros currently defined. Press F10 to display all currently loaded macros.
/H or /HISTORY	Displays previous commands with no line numbers; handy for creating batch files on-the-fly (use DOSKEY /H > BATNAME.BAT). Press F7 to display the command history with line numbers; for use with F9.
/INSERT or /OVERSTRIKE	Specifies insert or overwrite mode at installation. After installing, you can change to the opposite setting temporarily by pressing the INS (Insert) key.

Macros may use the following special characters:

\$1...\$9	Replaceable parameters. These are replaced by the first through ninth user-supplied parameters or arguments after the macro name on the command line.
\$*	Stands for <i>all parameters</i> and is replaced by all user-supplied text after the macro name on the command line; not limited to nine arguments.
\$t or \$T	Separates one command from the next; used to "chain" several commands together in a single macro. \$T stands for <i>terminate</i> , because it terminates or ends a command. You may chain as many commands together, using \$T, as will fit on the DOS 127-character command line.
\$g or \$G	Stands for <i>greater than</i> and is replaced by the redirected output symbol, ">," on the command line.
\$l or \$L	Stands for <i>less than</i> and is replaced by the redirected input symbol, "<," in a command.
\$b or \$B	Stands for <i>bar</i> and represents the piping symbol, " ."

—Eds.

for that command. Even then, it would be nice to be able to specify options such as /P or /W if needed. Replaceable parameters give you that flexibility by letting you change your command arguments from one use to the next. (Note: The table “Summary of DOSKEY Usage” briefly explains these parameters.)

Piping and Redirection in DOSKEY

Redirection and piping change the way commands give their output or take their input, and thus are useful in macros.

Piping and redirection symbols are like written punctuation marks that govern the flow of information. In a computer, this flow is between input devices, programs, and output devices. One of DOSKEY’s great benefits is its ability to automate redirection and piping. You no longer have to memorize how the symbols work or waste time typing the unfamiliar symbols. DOSKEY macros do the rote memorization and typing for you.

The main thing to realize is that the symbols must be disguised from COMMAND.COM, or you’ll get unexpected results. DOSKEY has its own symbols to keep COMMAND.COM from stepping in and misinterpreting a macro definition. Here is a summary of its piping and redirection symbols:

- **\$G:** Tells DOSKEY to redirect the output of a command to a file or device, instead of to its normal destination (usually the monitor); equivalent to the DOS greater than (>) sign, it is often used as \$G PRN to print a command’s output. When used to send output to a file, its output will overwrite and destroy an existing file of the same name, if one exists.
- **\$G\$G:** Also redirects output. It usually is used when redirecting to a filename in the command form, *command \$G\$G filename*. If a file of the same name already exists, \$G\$G will append the output to the existing file instead of overwriting it. If no file of the same name exists, \$G\$G will create one. The same as the double greater than (>>) symbols in DOS, it often is used to make entries to log files.
- **\$L:** Redirects input to a command; it tells a command to take its input from a specified program or input source, often a file or a filter program such as MORE or SORT. Useful for passing a known response such as “Y” or a carriage return to a command automatically, \$L is equivalent to the DOS less than (<) sign.
- **\$B:** Like the command-line piping (“|”) symbol, the “bar” \$B pipes a command’s output directly into another command, such as MORE or SORT. Where redirected input usually requires an input file of some sort, piping acts as a direct link. A temporary file is created, then immediately erased in the piping process.

The sidebar “DOS 6.0 Power Macros” provides examples of macros that use piping and redirection.

Easy Macro Saves and Reloads

One advantage macros have over batch files is that DOSKEY macros stay in memory. That makes them very fast, because they do not have to be read from a disk when you use them. But you need to be able to save your macros before you turn off the power, or you will lose them.

All information stored in RAM is volatile. Turning off power blanks the computer’s memory for macros just as it does for any other data currently in RAM. After you save your macros to disk, however, you can work with them like any other file, and they’ll be available for reloading the next day.

To save a complete list of your current macros in a file, you can redirect the output of the DOSKEY /M (or long-form DOSKEY /MACROS) command to a disk file. The following SAVEMACS macro will do it for you automatically:

```
SAVEMACS=DOSKEY /M $G DOSKEY.MAC
```

This macro redirects the list produced by DOSKEY /M to a macro file named DOSKEY.MAC in the current or default directory.

To save your macros in the same directory every time, add a path to the macro. For example, your DOS directory is probably on drive C. If you keep DOSKEY.COM there, you can modify SAVEMACS like this:

```
SAVEMACS=DOSKEY /M $G C:\DOS\DOSKEY.MAC
```

Modify the pathname to fit your own system so you’ll always know where the latest versions of all your macros are located.

Once you’ve defined your macro file, you need a standard technique for reloading your macros into memory when you start your computer. There are several ways to do this, but we recommend using one of the utility programs included on the disk that comes with our book. The program is MACLOAD, by Keith Graham. It is designed to load macros directly from a plain text file created by the SAVEMACS command.

You also can reload your macros using a batch file. First, save your macros in a file using SAVEMACS or an equivalent command. Next, load the resulting file into an editor that can produce plain ASCII files—the DOS Edit program will do fine—and insert the word *doskey* in front of each line. Name the file MACROS.BAT, or something like it. Now, when you enter the command MACROS, the MACROS.BAT file will load all your macros for you automatically.

If you decide to create macros in the MACROS.BAT batch file instead of creating them on-the-fly, write each line as a stand-alone command line. Start with the command DOSKEY, followed by the macro’s name, an equal sign (“=”), and the definition. ■

Edited by
Robert L. Hummel

TOOLS to Make You More PRODUCTIVE

TABLE OF CONTENTS

Page Description

MANAGING YOUR DISKS AND DIRECTORIES

54 Color-Coordinated Drives. Create batch programs that use color to tell you onto which drive you're logged.

54 A Smarter FORMAT. Wouldn't it be nice if FORMAT warned you that you were about to erase valuable data?

55 Improving on MKDIR. A slick batch program to ease your directory navigation.

56 A Quick Trip Home. You can go home again, using this batch program to return quickly to your favorite directory.

57 Pushing and Popping Directories. With these programs, you won't need to leave a trail of bread crumbs to find your way back home on your hard disk.

58 Better Directory Assistance. RUN.BAT lets you find and run programs stored anywhere on your system.

59 Controlling Your PATH. Add and remove directories easily with this pair of batch programs.

MANAGING YOUR FILES

59 Viewing Files. A kinder alternative to the TYPE command, VIEW.BAT automatically pauses at the end of each page.

60 Swapping Files with QBasic. Put programming power to work moving your files between directories.

60 A Verbose XCOPY. If you use this batch file, XCOPY will gladly tell you whether its copy operation was successful.

61 COPY Minus the Clobber. Add overwrite protection to any version of DOS.

63 Easier Backups. This clever batch file lets you use XCOPY to back up files across several disks.

63 Checking Up on COPY. If that file absolutely, positively has to be where you put it, you need this copy-verification batch program.

64 Power Deleting. Don't wait for the next version of DOS. Get a DEL command that accepts multiple file specifications now.

64 Clean Your Disk with QBasic. Use this program to sweep your hard disk clean of pesky backup files.

65 Making FIND Logical. This nifty batch file lets the FIND command perform Boolean searches on your files.

66 Filtering Out Enter. A short Debug utility is all you need to remove extra characters from text files.

66 Combining Files. This program gathers related files into an easy-to-manage bundle.

67 Making Unique Filenames. Don't wrack your brain. Let this QBasic program do the work for you.

68 A Poor Man's Data Compression. This helper program for PKZIP saves you time and disk space.

70 Comparing Directories. Is that backup file exactly the same as the one on your hard disk? Find out in a flash.

HARDWARE HELPERS

71 Put More Lines on Your Screen. If you're tired of watching directory listings scroll off the top of your screen, try this.

71 A Basic Screen Saver. This mesmerizing display will prevent pixel burn-in and discourage curious snoops from tinkering with your computer.

72 Colorful Start-Ups. Bring some color into your life with this handy screen utility.

72 Controlling Your Cursor. Give yourself something DOS and ANSI.SYS lack—the ability to turn your cursor on and off.

74 I Brake for Games. PC too fast? Get a grip on it with this time-wasting utility.

74 Check the Date. Give your batch files the ability to tell time and make informed decisions.

76 Swapping Printer Ports. Stop hunting for those small screwdrivers! Use software to swap your printers.

76 A PC Typewriter. Here's a simple method for using your printer like a typewriter.

78 Function Keys and ANSI.SYS. Unlock the power of your keyboard by creating a bundle of macro keys.

78 Toggling Num Lock Remotely. Forget fumbling for keys. Use these short utilities to turn Num Lock on and off.

This collection of batch programs, Debug scripts, and clever command techniques takes the work off your shoulders and puts it back on DOS, where it belongs.

Lately, it seems that everywhere you turn, someone is criticizing DOS. Some maintain that it's difficult to learn and use. Others claim it's a crude and primitive environment that lacks the glamour of Windows. A few even go so far as to say that it's about time to bury DOS for good, because no new applications are being written for it. However,

SAVE TYPING TIME:**HOW TO GET THE LISTINGS ON DISK**

Every listing in the *Maximum DOS Tool Kit* is available on the Companion Disk. For ordering information, turn to page 79. Or, you can download the listings in a single ZIPped file by calling the *DOS Resource Guide* BBS at 1-603-924-3181. Follow the on-screen prompts for directions. (There is no connect time charge—you pay only for the phone call.)

the reports of DOS's demise are greatly exaggerated.

DOS continues to reign as the most popular PC operating system in the world. Why? Because no other operating system is as well understood, as well documented, or as easy to enhance as DOS.

There are many ways to give DOS a boost. You might use a commercial application such as The Norton Utilities to improve your system's performance. Or you might revel in developing intelligent batch programs that automate your daily computing chores. But often, the cheapest and most direct route to more efficient computing is to borrow ideas and programs from other users.

With that in mind, I've assembled a collection of DOS-improvement programs culled from the pages of *DOS Resource Guide*. You'll find a mixed bag of short batch listings, longer batch programs, and QBasic and assembly language listings—all designed to help you accelerate DOS to light speed. Think of it as a home-made tool kit—small but useful utilities that you can assemble yourself in a few hours.

To help you locate the help you need fast, the programs are organized under three section headings: Managing Your Disks and Directories, Managing Your Files, and Hardware Helpers. Whether you want to goof-proof your FORMAT command, find a better way to search your files, or turn off that pesky Num Lock key, you're sure to find the solution here.

MANAGING YOUR DISKS AND DIRECTORIES

In this day of monster hard disks, it's difficult to imagine that the original IBM PC was designed to work without any disk drives at all. IBM believed that home users would eschew the pricey 160K single-sided floppy drives and opt instead for cheaper cassette tape recorders. Fortunately, good sense prevailed, and the cassette option disappeared with the introduction of the PC/XT.

Over the years, our dependence on disks has increased

dramatically. A hard disk no longer is considered a luxury, and the bigger it is, the faster we seem to fill it up. Unfortunately, the disk-management tools that DOS provides can't keep up with burgeoning disks and myriad directories. Try finding a single file on a crowded hard disk or maneuvering around a complex subdirectory structure. Even the most patient among us grow frustrated.

Don't panic! These eight programs can help you get a grip on your hard disk problems.

COLOR-COORDINATED DRIVES

If you have a color monitor and more than one hard drive, you may find it useful to create a batch program that sets a different foreground and background color for each drive. When you execute the batch program, it changes you to the drive you specify and alters the screen colors automatically. (Before you begin, make sure CONFIG.SYS includes a DEVICE=ANSI.SYS command.)

If, for example, you want the screen to display black text on a red background when you are on drive D, you may use the following program, called D.BAT:

```
@ECHO OFF
CLS
D:
PROMPT $E[41;30m$P$G
```

To change to drive D, just type D at the DOS prompt.

If you want drive C to display different screen colors, create a file called C.BAT and execute it to switch to drive C. This file might, for instance, contain a PROMPT statement such as the following, which restores the display to DOS's default colors:

```
PROMPT $E[0m$P$G
```

—Kay Yarborough Nelson

A SMARTER FORMAT

One of FORMAT's problems is that it never tells you whether a disk already is formatted and contains data. If you have DOS 5 or later, this shortcoming is annoying, but hardly catastrophic. With these DOS versions, FORMAT follows "safe formatting" procedures whenever it formats a disk (unless you use the unconditional format switch, /U). If you foul up, you may use UNFORMAT to recover lost data. But wouldn't it be nice if

you could rely on a formatting utility that blurts out, "Hey! That floppy disk is formatted already!" and then asks for confirmation before reformatting it?

Well, I have one you can use. To take advantage of it, you must create a trio of short utilities and a pair of text files. The executable programs GETDRIVE.COM, DISKIN.COM, and ASK.COM have different functions, but they are alike in one respect: You create them as text files and convert them to executable programs using DOS's Debug program. To create GETDRIVE.COM, for instance, simply type its script file, GETDRIVE.SCR, into an editor or word processor that saves files in text (ASCII) format. To create GETDRIVE.COM, feed the commands in GETDRIVE.SCR into Debug by typing the following input redirection command at the DOS prompt:

```
DEBUG < GETDRIVE.SCR
```

Do the same for the DISKIN.SCR script and ASK.SCR.

Next, to create ENTERN, a text file containing three key presses—an Enter, an *N*, and another Enter—type the following commands at the DOS prompt, pressing the Enter key after each line:

```
COPY CON C:\BATCH\ENTERN
N ^Z
```

Press Ctrl-Z or F6 to end the file and make the sequence ^Z appear on screen. Finally, type in the listing. When you're done, save the file as FORMAT.BAT.

Place the COM files in your C:\BATCH directory—or in a subdirectory mentioned in your PC's search path.

Otherwise FORMAT.BAT won't be able to find them.

Reprogramming FORMAT

To keep FORMAT.BAT from trying to call itself, or to prevent DOS's FORMAT command from executing instead of the batch program, rename the latter as FORMAT!.COM. (You'll note that my version of FORMAT.BAT refers to the DOS command by this new name.) If you're skittish about renaming a DOS command, name FORMAT.BAT something else, and change line 32 to read :FORMAT.

FORMAT.BAT does several things the FORMAT command should do: It asks which drive contains the floppy disk you want to format, tells you if the disk is in the drive and whether it is formatted, and formats the disk for you.

—Dan Gookin

IMPROVING ON MKDIR

If you're reorganizing your hard disk, making a directory and then switching into it gets old fast. Usually you create the directory, forget that you need to switch to it, and end up copying files to the wrong directory. Ugh!

ND.BAT (ND stands for *new directory*) neatly solves the problem. The listing contains just three lines:

```
@ECHO OFF
MD %1
CD %1
```

Use the program as you would the MKDIR (MD) command, typing its name followed by the name of the di-

Use GETDRIVE.SCR to create the executable program GETDRIVE.COM.

```
N GETDRIVE.COM
E100 BA 2B 01 B4 09 CD 21 B4
E108 08 CD 21 24 5F 3C 41 74
E110 04 3C 42 75 F2 50 8A D0
E118 B4 02 CD 21 B2 0D CD 21
E120 B2 0A CD 21 58 2C 41 B4
E128 4C CD 21
E12B "Which drive (A/B)?$"
```

```
RCX
3E
W
Q
```

End

Use DISKIN.SCR to create the executable program DISKIN.COM.

```
N DISKIN.COM
E100 A0 80 00 0A C0 74 09 A0
E108 82 00 24 5F 2C 41 EB 02
E110 B0 00 B9 01 00 BA 00 00
E118 BB 37 01 CD 25 73 12 80
E120 FC 1F 73 09 80 FC 08 74
E128 04 B0 01 EB 06 B0 02 EB
E130 02 B0 00 B4 4C CD 21
```

```
RCX
37
W
Q
```

End

Use ASK.SCR to create the executable program ASK.COM.

```
N ASK.COM
E100 A0 80 00 3C 00 74 26 98
E108 8B C8 BB 82 00 E8 33 00
E110 B4 08 CD 21 24 5F 3C 59
E118 74 0E 3C 4E 75 F2 8A D0
E120 E8 0F 00 B8 01 4C CD 21
E128 8A D0 E8 05 00 B8 00 4C
E130 CD 21 B4 02 CD 21 B2 0D
E138 B4 02 CD 21 B2 0A B4 02
E140 CD 21 C3 8A 17 80 FA 0D
E148 74 07 B4 02 CD 21 43 E2
E150 F2 B2 20 B4 02 CD 21 C3
```

```
RCX
58
W
Q
```

End

FORMAT.BAT adds several important safeguards when using DOS's **FORMAT** command. Be sure to rename DOS's **FORMAT.COM** to **FORMAT!.COM** before running the batch program.

```
@ECHO OFF
ECHO Killer Disk Formatter
ECHO.
ECHO You are about to format a floppy disk.
GETDRIVE
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 SET DRIVE=B:
IF ERRORLEVEL 0 IF NOT ERRORLEVEL 1 SET DRIVE=A:

:START
DISKIN %DRIVE%
IF ERRORLEVEL 2 GOTO MISSING
IF ERRORLEVEL 1 GOTO UNFORMAT

REM Floppy disk is formatted.
ECHO That disk is formatted!
ASK Continue?
IF ERRORLEVEL 0 IF NOT ERRORLEVEL 1 GOTO FORMAT
ASK Format another floppy disk?
IF ERRORLEVEL 1 GOTO END
GOTO START

:MISSING
ECHO There is no floppy disk in that drive!
```

```
ECHO Put a floppy disk into drive %DRIVE% and
PAUSE
GOTO START

:UNFORMAT
ECHO Unformatted floppy disk detected.
:FORMAT

ECHO Formatting the floppy disk...
C:\DOS\FORMAT! %DRIVE% /V:NONE %1 %2 %3 %4
<C:\BATCH\ENTERN
ECHO.
ECHO Format successful!
ASK Format another floppy disk?
IF ERRORLEVEL 1 GOTO END
ECHO Remove the floppy disk in drive %DRIVE%
ECHO and replace it with a new disk.
PAUSE
GOTO START

:END
SET DRIVE=
```

End

rectory you want to create. This command, for instance, creates and then switches you to \NEW:

```
ND \NEW
```

—David Oatley

A QUICK TRIP HOME

If your hard disk has a complex directory structure nested several levels deep, you'll probably recognize this situation. You need to leave one subdirectory to check something in another. Later, you want to switch back quickly, but DOS doesn't provide any shortcuts for accomplishing this task. Why not build a set of small batch programs to do it for you?

The batch program **RETURN.BAT** marks the current directory as home and builds a second batch program to return you to that home directory. The next time you run this batch program, it creates a new home directory and a new batch program to return you there.

Before using **RETURN.BAT**, you must create a file called **RETURN** by typing the following commands at the DOS prompt, pressing the Enter key after each line:

```
COPY CON RETURN
CD ^Z
```

Press Ctrl-Z or F6 to end the file and make the sequence ^Z appear on screen. For **RETURN.BAT** and **RETURN** to work, you must place them in a subdirectory mentioned in **AUTOEXEC.BAT**'s **PATH** statement. As written, **RETURN.BAT** uses the subdirectory **C:\BATCH**. If you prefer, you can substitute another drive or directory name.

To use **RETURN.BAT**, type **RETURN** while you are in the

RETURN.BAT creates a batch program called **RETURNTO.BAT** to store the name of the current subdirectory and let you easily change back to it.

```
@ECHO OFF
REM RETURN.BAT
REM Returns you to a subdirectory
IF (%1)==(/?) GOTO HELP
IF (%1)==(?) GOTO HELP
COPY C:\BATCH\RETURN C:\BATCH\RETURNTO.BAT
CD >> C:\BATCH\RETURNTO.BAT
GOTO END
:HELP
ECHO This batch program creates another
ECHO batch program called RETURNTO.BAT
ECHO that will return you to your
ECHO previous subdirectory automatically.
GOTO END
:END_HELP
:END
```

End

subdirectory to which you want to return. Type RETURNTO when you are in another subdirectory and you want to return to the subdirectory you were in when you invoked RETURN.BAT. Note that RETURN can't switch you between directories on different drives.

PUSH.BAT saves the names of as many as eight recently visited directories, permitting you to recall those names later.

```
@ECHO OFF
REM PUSH.BAT
IF "%1"==" " GOTO NEW
IF "%9"==" " GOTO PUSHIT
ECHO Stack is full.
GOTO END
:PUSHIT
ECHO Pushing...
@ECHO OFF
SET POPER=POP
ECHO %0 %1 %HOMEDIR% %2 %3 %4 %5 %6 %7 %8 >
C:\BATCH\STK1.BAT
ECHO %POPER% %1 %HOMEDIR% %2 %3 %4 %5 %6 %7 %8 >
C:\BATCH\STK2.BAT
GOTO END
:NEW
DIRECTOR
:END
```

End

POP.BAT lets you quickly return to the eight directories most recently saved by PUSH.BAT.

```
@ECHO OFF
REM POP.BAT
IF "%1"==" " GOTO NEW
FOR %%X IN (1,8) DO IF %%X==%1 GOTO SHOWEM
IF NOT "%2"==" " GOTO POPIT
ECHO Stack is empty.
GOTO END
:SHOWEM
TYPE C:\BATCH\STK2.BAT
GOTO END
:POPIT
ECHO.
ECHO Popping...
@ECHO OFF
SET POPER=PUSH
ECHO %0 %1 %3 %4 %5 %6 %7 %8 %9 > C:\BATCH\STK2.BAT
ECHO %POPER% %1 %3 %4 %5 %6 %7 %8 %9 >
C:\BATCH\STK1.BAT
C:\BATCH\STK1.BAT
%2\
CD %2
GOTO END
:NEW
C:\BATCH\STK2.BAT
:END
```

End

PUSHING AND POPPING DIRECTORIES

Here's a group of batch programs that let you create a "stack" of previously used directories as you move around your hard drive, then quickly and easily return to those directories. This group of files is particularly useful if you jump around a lot.

The first file, PUSH.BAT, saves as many as eight directories. When you're in a directory to which you may want to return later, type PUSH at the DOS prompt. PUSH.BAT adds that directory to its stack. If you try to save more than eight directories, PUSH.BAT displays a "Stack is full" message.

To return to a previously "pushed" directory, simply type POP at the DOS prompt. POP.BAT finds the most recent entry placed on the stack by PUSH.BAT and switches you to that directory. Each successive use of POP.BAT switches you to the next most-recent entry on the stack until no more entries remain, at which point you receive a "Stack is empty" message. To look at the list of directories on the stack at any time, type POP L.

DIRECTOR.BAT is called by PUSH.BAT to determine the name of the current directory, so that it can be saved to the stack. STK1.BAT and STK2.BAT are batch programs

DIRECTOR.BAT is used by PUSH.BAT to capture the name of the current directory.

```
REM DIRECTOR.BAT
IF "%1"=="of" GOTO ICALLME
CTTY NUL
DIR DUMMY.$$$ | FIND "Directory of " > TEMP.BAT
CTTY CON
TEMP
GOTO END
:ICALLME
SET HOMEDIR=%2
DEL TEMP.BAT
C:\BATCH\STK1.BAT
:END
```

End

STK1.BAT is one of two files used by PUSH.BAT and POP.BAT to store a list or "stack" of directories. Both PUSH.BAT and POP.BAT modify this file, but this is its initial form.

```
REM STK1.BAT
PUSH *
```

End

STK2.BAT is the second file used to store directory information. Like STK1.BAT, it is modified by PUSH.BAT and POP.BAT.

```
REM STK2.BAT
POP *
```

End

containing the directory stack information. Both stack files are modified when you run PUSH.BAT and POP.BAT; don't be surprised if you examine them and find that their contents no longer match the lines listed here.

You must place PUSH.BAT, POP.BAT, and DIRECTOR.BAT in a directory listed in your PATH command. STK1.BAT and STK2.BAT may reside in any directory you choose, as long as you modify PUSH.BAT, POP.BAT, and DIRECTOR.BAT accordingly. I put STK1.BAT and STK2.BAT in my C:\BATCH directory. If you use a different directory, you must change the pathname provided in lines 11 and 12 of PUSH.BAT; lines 9, 16, 17, and 22 of POP.BAT; and line 11 of DIRECTOR.BAT.

—Herbert Feldman

BETTER DIRECTORY ASSISTANCE

I sometimes find myself at the DOS prompt, where I'm forced to maneuver manually around my computer's disk drives. I created a batch program called RUN.BAT,

which lets me run applications located on other drives. It also accepts command-line parameters, which it passes to the application when it is started.

Using RUN.BAT couldn't be simpler. The following command, for example, executes QBasic (located on drive D) and loads a program called MYFILE.BAS.

```
RUN D: QBASIC + MYFILE.BAS
```

The program checks to ensure that you specify a parameter and don't provide a command-line parameter without specifying an application. If a drive change is required, RUN.BAT changes to the correct drive, then shifts the command-line parameters so that other sections of the batch program operate correctly. The remainder of RUN.BAT searches for the application and runs it.

To pass command-line parameters to the specified program, RUN.BAT must have a means of distinguishing between parameters and directory or program names. My solution is to precede command-line parameters with a plus sign (+). The first IF statement following the label LOOP checks to ensure that the first parameter is not a plus. If it is, an error exists, because when you invoke

RUN.BAT lets you use a single command line to run a program on any drive or in any directory.

```
@ECHO OFF
IF "%1"==" " GOTO ERROR
IF "%1"=="+" GOTO ERROR1
CD\
IF "%2"==" " GOTO LOOP
FOR %%P IN (A: a:) DO IF "%1"=="%%P" GOTO DRIVE
FOR %%P IN (B: b:) DO IF "%1"=="%%P" GOTO DRIVE
FOR %%P IN (C: c:) DO IF "%1"=="%%P" GOTO DRIVE
FOR %%P IN (D: d:) DO IF "%1"=="%%P" GOTO DRIVE
FOR %%P IN (N: n:) DO IF "%1"=="%%P" GOTO DRIVE
GOTO LOOP
:DRIVE
  %1
  SHIFT
  CD\
:LOOP
  IF "%1"=="+" GOTO ERROR1
  IF "%2"==" " GOTO DONE
  IF "%2"=="+" GOTO DONE
  IF EXIST %1\NUL GOTO CHGDR
  GOTO ERROR3
:CHGDR
  CD %1
  SHIFT
  GOTO LOOP
:DONE
  IF EXIST %1.COM GOTO RUN
  IF EXIST %1.EXE GOTO RUN
  IF EXIST %1.BAT GOTO RUN
  IF EXIST %1\NUL GOTO CHGDR1
:CHGDR1
  CD %1
  IF EXIST %1.COM GOTO RUN
  IF EXIST %1.EXE GOTO RUN
  IF EXIST %1.BAT GOTO RUN
  SHIFT
```

```
IF "%1"==" " GOTO ERROR2
:RUN
  IF "%2"=="+" SET CMD=%1 %3
  IF "%2"==" " SET CMD=%1
  %CMD%
  GOTO END
:ERROR
  CLS
  ECHO Syntax:
  ECHO RUN [drive] [directory] [program] [+
    parameters]
  GOTO ENDERROR
:ERROR1
  CLS
  ECHO You may type a plus only if you
  ECHO follow it with command-line parameters.
  GOTO ENDERROR
:ERROR2
  CLS
  ECHO You typed the name of an executable file
  ECHO that doesn't exist.
  GOTO ENDERROR
:ERROR3
  CLS
  ECHO You selected an invalid
  ECHO directory.
:ENDERROR
  ECHO.
  PAUSE
:END
  SET CMD=
  C:
  CD\
  CLS
```

End

RUN.BAT, you must follow a plus with the name of an executable file.

—John F. Lohmann

CONTROLLING YOUR PATH

Normally, if you want to change your path, you type the new path at the DOS prompt or modify your AUTOEXEC.BAT's PATH statement. But that's hardly efficient. A better method for handling this chore involves creating a simple pair of batch programs called NEWPATH.BAT and OLDPATH.BAT.

NEWPATH.BAT adds a directory to the current path. If, for example, you want to add a directory called C:\WP51 to your search path, you must type:

```
NEWPATH C:\WP51
```

Before it alters your search path, NEWPATH.BAT saves

NEWPATH.BAT lets you quickly add a directory to your existing search path.

```
@ECHO OFF
REM NEWPATH.BAT
IF "%1"==" " GOTO NODIR
SET OLD=%PATH%
PATH=%PATH%;%1
ECHO The new path is %PATH%.
ECHO Type OLDPATH to restore the original path.
GOTO END
:NODIR
ECHO You didn't specify the name of a directory
ECHO to add to the path. Please try again, using
ECHO the proper syntax: NEWPATH directory.
:END
```

End

OLDPATH.BAT, the companion file to NEWPATH.BAT, provides a way to restore your old path when you no longer need your temporary path.

```
@ECHO OFF
REM OLDPATH.BAT
IF "%OLD"==" " GOTO NOOLD
PATH=%OLD%
SET OLD=
ECHO The original path has been restored.
GOTO END
:NOOLD
ECHO The environment doesn't contain a variable
ECHO called OLD. There is no old path to restore!
:END
```

End

the old path in an environment variable named OLD. Then, the string C:\WP51 is added to the search path. Next, an ECHO command displays the environment variable's values on the screen. A second ECHO statement advises you to type OLDPATH to restore your path to its original state.

OLDPATH.BAT first checks to see if the variable OLD exists. If it doesn't find OLD, the batch program tells you that there is no path to restore, and the program ends. If OLD exists, the batch program uses the string stored in that variable to restore the old path. OLD then is removed from the environment.

MANAGING YOUR FILES

In the good old days, you could keep your entire software library on a half-dozen 360K disks. Today, it's common for a single large application to install more than 100 files to your hard disk. No one wants to return to the days before hard drives, but it would be nice if you could make life simpler.

If dealing with a bad case of file overload has made you frazzled, try a few of these hot ideas. The collection of programs presented here offers answers to many of your file-management problems. You'll find programs to help you with nearly every file operation including viewing, naming, copying, deleting and undeleting, and compressing.

VIEWING FILES

If you frequently use the TYPE command to view text files stored in standard ASCII format, the following batch program, which you may name VIEW.BAT, will simplify your life:

```
CLS
TYPE %1 | MORE
CLS
```

Put VIEW.BAT to work by invoking the batch program and specifying the name of the file you want to display. For example, to view the contents of a file called MYTEXT.ASC—one screen at a time—type:

```
VIEW MYTEXT.ASC
```

DOS will pause at the end of each full screen of text,

giving you the opportunity to read what is on the screen. Pressing any key tells DOS to display the next section of your file.

—Jack Nimersheim

SWAPPING FILES WITH QBasic

One of the most frustrating things about DOS used to be that it wouldn't let you move files between directories. The addition of the MOVE command in DOS 6.x fixed this shortcoming, but users of earlier versions of DOS are still stuck. If you don't have DOS 6.x but do have a copy of QBasic on your PC, you can use my short batch program, SWAP.BAT to produce the equivalent of the MOVE command. If you do have DOS 6.x, you may still benefit from using SWAP.BAT, because unlike MOVE it also lets you swap the names of two files that reside in different directories.

SWAP.BAT creates the QBasic program MYSWAP.BAS by redirecting a series of ECHO statements to a disk file. The result is a customized five-line Basic listing that looks something like this:

```
ON ERROR GOTO 5:NAME "%1" AS "MYSWAP.TMP"
ON ERROR GOTO 4:NAME "%2" AS "%1"
NAME "MYSWAP.TMP" AS "%2":SYSTEM
NAME "MYSWAP.TMP" AS "%1"
PRINT "ERROR: MYSWAP %1 %2 FAILED.":SYSTEM
```

When the file is created on the disk, however, the %1

and %2 parameters are replaced with the command-line arguments that you supplied on the command line.

Although SWAP.BAT is sophisticated, it has its share of limitations. First of all, you can't use the wildcards * and ? with the program. Secondly, the files that you swap must be on the same drive; as a result, you should exercise caution when using SWAP.BAT with joined directories.

—David Phelps

A VERBOSE XCOPY

The XCOPY command offers a number of improvements over the plain-vanilla COPY command. One benefit you may not be aware of, however, is that XCOPY returns an exit code that indicates the result of the COPY operation. If you execute XCOPY from the command line, the exit code is lost.

The VCOPY.BAT program uses the XCOPY command to transfer all files that reside in the source directory to the target directory. After XCOPY terminates, the batch program reports any errors that occur or tells you the operation was a success.

You may type in VCOPY.BAT using Edit, Edlin, or any word processor that saves files in ASCII format. Save the batch program as VCOPY.BAT, then type:

```
VCOPY source_file target_file
```

(Be sure to specify pathnames when appropriate.)

To copy all files in the directory C:\WP\LETTERS to a

SWAP.BAT uses QBasic to overcome DOS's inability to rename, and therefore swap, files across directories.

```
@ECHO OFF
IF !%2==! GOTO HELP
IF NOT EXIST %1 GOTO ERR_1
IF NOT EXIST %2 GOTO ERR_2
IF EXIST MYSWAP.TMP GOTO ERR_3
REM If you have a RAM disk, add a path to the ECHO
REM statements to store MYSWAP.BAS on that drive.
ECHO 1 ON ERROR GOTO 5:NAME "%1" AS "MYSWAP.TMP" >
MYSWAP.BAS
ECHO 2 ON ERROR GOTO 4:NAME "%2" AS "%1" >> MYSWAP.BAS
ECHO 3 NAME "MYSWAP.TMP" AS "%2":SYSTEM >> MYSWAP.BAS
ECHO 4 NAME "MYSWAP.TMP" AS "%1" >> MYSWAP.BAS
ECHO 5 PRINT "ERROR: MYSWAP %1 %2 failed.":SYSTEM >>
MYSWAP.BAS
REM If you have GW-Basic or BasicA, change
REM the following command accordingly:
QBASIC /RUN MYSWAP.BAS
DEL MYSWAP.BAS
DEL MYSWAP.TMP
GOTO END
```

```
:ERR_1
ECHO "%1" not found.
GOTO END

:ERR_2
ECHO "%2" not found.
GOTO END

:ERR_3
ECHO The temporary file MYSWAP.TMP already exists.
ECHO Delete or rename MYSWAP.TMP.
GOTO END

:HELP
ECHO The program uses the following syntax:
ECHO MYSWAP [path\]file1 [path\]file2

:END
```

End

VCOPY.BAT uses **IF ERRORLEVEL** and **XCOPY** exit codes to verify the success of transferring files from a source directory (%1) to a target directory (%2).

```
@ECHO OFF
IF "%1" == "" GOTO NO_SRC
IF "%2" == "" GOTO NO_DEST
XCOPY %1 %2
IF ERRORLEVEL 4 GOTO CATCH_ALL
IF ERRORLEVEL 2 GOTO CTRL_C
IF ERRORLEVEL 1 GOTO NO_FILES
IF ERRORLEVEL 0 GOTO SUCCESS

:CATCH_ALL
ECHO One of the following errors has occurred:
ECHO -----
ECHO 1. There is not enough memory to copy the
    files.
ECHO 2. You specified a drive that doesn't exist.
ECHO 3. You mistyped the source/destination
    pathname.
GOTO END_IT

:SUCCESS
ECHO Files in %1 were copied to %2
GOTO END_IT

:NO_FILES
ECHO The source directory %1 either doesn't exist
ECHO or contains no files to copy.
GOTO END_IT

:NO_SRC
ECHO You didn't specify a source directory.
GOTO END_IT

:NO_DEST
ECHO You didn't specify a destination directory.
GOTO END_IT

:CTRL_C
ECHO The program stopped because you pressed Ctrl-C.

:END_IT
```

End

floppy disk in drive A, type:

```
VCOPY C:\PW\LETTERS A:
```

Before you start **VCOPY.BAT**, make sure the directory in which you store **XCOPY.EXE** is included in the **PATH** statement of your **AUTOEXEC.BAT** file. Normally, **XCOPY.EXE** resides in your **C:\DOS**, which usually is included in your **PATH** statement.

—John Wolfskill

COPY MINUS THE CLOBBER

You don't have to upgrade to MS-DOS 6.2 to find a way to avoid clobbering files during a copy operation. My short batch file, **CP.BAT**, includes an **IF EXIST** test that provides a means of checking for the existence of files before **COPY** sets to work. If **CP.BAT** discovers that a file of the same name already exists, it displays a warning and stops copying.

To use the program, type in the listing, using an editor that saves files in ASCII format. When you invoke the program, be sure to supply a complete destination path and filename.

To make **CP.BAT** into a batch program that moves a file to a new location—first copying a file and later deleting the original file—insert the command **DEL %1** before the final **GOTO END** statement. That portion of the listing will then look like this:

```
ECHO FILES COPIED
DEL %1
GOTO END
:WARNING
```

You may name the file **MV.BAT** and use it as your anti-clobber, file-moving batch program if you like.

But be careful with either program! In most circum-

CP.BAT is a **COPY** alternative that gives older DOS versions some of the file-overwrite protection available in DOS 6.2.

```
@ECHO OFF
REM COPY FILES FROM %1 TO %2
REM Test parameters:
IF %1NOTHING==NOTHING GOTO WARNING
IF %2NOTHING==NOTHING GOTO WARNING
REM Test for existence of duplicate files:
IF NOT EXIST %2 GOTO CP01
ECHO %2 already exists!
GOTO END
:CP01
REM Test for subdirectory\file(s):
IF NOT EXIST %2\%1 GOTO CP02
ECHO %2\%1 already exists!
GOTO END
:CP02
ECHO Copying File(s)...
COPY %1 %2 >NUL
ECHO Files copied.
GOTO END
:WARNING
ECHO When you invoke CP.BAT, you must specify
ECHO a source and destination filename.
:END
```

End

SAVENEW.BAT uses XCOPY's exit codes to let you back up files onto multiple disks.

```

@ECHO OFF
REM SAVENEW.BAT
REM Syntax: SAVENEW [x] [y]
REM Both x and y are optional. If they are used,
REM they are the destination and source drives.
REM This program assumes that you want to save files
REM from a hard disk to a floppy disk. The order of
REM x and y is unimportant. If the source isn't
REM specified, the program assumes drive C. If the
REM destination isn't specified, the program
REM prompts for the correct drive.
REM CLS
ECHO SAVENEW backs up all new and altered files from
ECHO your hard disk to a floppy disk. It formats the
ECHO floppy disks for you, if you wish.
ECHO Press Ctrl-C to stop or
PAUSE
SET SOURCE=C
SET DEST=
SET FMT=
IF "%1"==" " GOTO GETDEST
FOR %%x IN (A a B b) DO IF %%x==%1 SET DEST=%1
IF "%DEST%"==" " SET SOURCE=%1
IF "%2"==" " GOTO GETDEST
FOR %%x IN (A a B b) DO IF %%x==%2 SET DEST=%2
IF NOT "%DEST%"=="%2" SET SOURCE=%2
:GETDEST
IF NOT "%DEST%"==" " GOTO STARTCOPY
REM CLS
ECHO Enter the destination for the files as either
ECHO A for drive A or B for drive B
CHOICE /CAB
IF ERRORLEVEL 2 SET DEST=B
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 SET DEST=A
:STARTCOPY
REM CLS
ECHO Ready to back up all new and altered files from
ECHO drive %SOURCE%.
ECHO The files will be saved on floppy disks in
ECHO drive %DEST%.
ECHO.
ECHO Press Ctrl-C to stop or
PAUSE
:GETDISK
REM CLS
ECHO Put a floppy disk into drive %DEST%.
PAUSE
ECHO Do you want to format this disk?
CHOICE /CYN
IF ERRORLEVEL 2 GOTO DOCOPY
IF NOT "%FMT%"==" " GOTO DOFORMAT
ECHO.
ECHO.
ECHO Enter 1 for a 5.25-inch, 360K disk
ECHO Enter 2 for a 5.25-inch, 1.2MB disk
ECHO Enter 3 for a 3.5-inch, 720K disk

```

```

ECHO Enter 4 for a 3.5-inch, 1.4MB disk
CHOICE /C1234
IF ERRORLEVEL 4 SET FMT=/F:1440
IF ERRORLEVEL 3 IF NOT ERRORLEVEL 4 SET FMT=/F:720
IF ERRORLEVEL 2 IF NOT ERRORLEVEL 3 SET FMT=/F:1200
IF ERRORLEVEL 1 IF NOT ERRORLEVEL 2 SET FMT=/F:360
:DOFORMAT
FORMAT %DEST%: %FMT%
IF NOT ERRORLEVEL 1 GOTO DOCOPY
REM CLS
ECHO FORMAT failed to format the disk in drive %DEST%.
ECHO The command that failed was:
ECHO.
ECHO     FORMAT %DEST%: %FMT%
GOTO END
:DOCOPY
XCOPY %SOURCE%:\*.* %DEST%: /S /M
IF ERRORLEVEL 6 GOTO UNKNOWNERROR
IF ERRORLEVEL 5 GOTO WRITEERROR
IF ERRORLEVEL 4 GOTO GETDISK
IF ERRORLEVEL 3 GOTO UNKNOWNERROR
IF ERRORLEVEL 2 GOTO USERBREAK
IF ERRORLEVEL 1 GOTO NOFILES
:SUCCESS
REM CLS
ECHO All new and altered files on drive %SOURCE%
ECHO were saved on drive %DEST%.
GOTO END
:UNKNOWNERROR
ECHO.
ECHO XCOPY reported an unknown error while
ECHO executing the command.
ECHO.
ECHO XCOPY %SOURCE%:\*.* %DEST%: /S /M
GOTO END
:WRITEERROR
ECHO.
ECHO XCOPY reported a write error on drive %DEST%.
ECHO.
ECHO Do you want to continue with a new disk?
CHOICE /CYN
IF ERRORLEVEL 2 GOTO GETDISK
GOTO END
:USERBREAK
ECHO.
ECHO XCOPY was interrupted at your request.
GOTO END
:NOFILES
ECHO.
ECHO XCOPY did not find any files on drive %SOURCE%.
:END
ECHO.
ECHO The SAVENEW program is now ending.

```

End

stances, CP.BAT successfully copies files, but in a few situations it can be fooled and may end up clobbering a file already on disk. (You can create a batch file that checks for these conditions, but that makes the program long and extremely slow.)

—Dan Gookin

EASIER BACKUPS

Every time a program creates or updates a data file, EDOS sets the file's archive attribute to warn you that the file should be backed up or archived. If you use XCOPY with its /M option, it copies only files that have the archive bit set and, as each file is copied successfully, turns off the archive bit. This switch is used most often in an automated backup routine. If you want to save all of the new files on drive C, turn off the archive bit of each, and store the files on a floppy in drive A, using this command:

```
XCOPY C:\*.* A: /M /S
```

(The /s switch tells XCOPY to search through all subdirectories for files to copy.)

That command, by itself, does the trick if the files fit on one disk. If they don't, you must use XCOPY's exit codes in a batch program to save files on multiple disks. XCOPY returns an exit code of 4 if the XCOPY command line contains a syntax error, if you ask XCOPY to copy from or to an invalid disk name, if your computer has insufficient memory, or—most importantly—if the destination disk is full. Using that information, you can outline the logic for a backup batch program this way:

- Prompt the user to insert a fresh disk in drive A.
- Execute XCOPY.
- If the exit code is 4, go to the first step.
- If the exit code is zero or 1, end.
- Report any other error that occurred.

This logic also looks for other error conditions and reports them, so that you know if XCOPY stopped because of an error rather than because all the changed files were saved.

To be really useful, the batch program also should offer to format each disk that the user inserts and warn when the user is trying to back up files onto a disk that already contains files. And it should let the user back up files onto a disk in either floppy drive. A final backup batch program with all of these enhancements is shown in SAVENEW.BAT.

—Hardin Brothers

CHECKING UP ON COPY

You might expect the /v (verify) switch of the DOS COPY command to verify the contents of the files it copies. However, /v merely verifies that the target area of the destination disk can be read without an error. The /v switch does not make a byte-by-byte comparison of the files.

To ensure that your source and destination files are exactly alike, you must use one of DOS's file-comparison commands—FC or COMP—after you copy the files from one disk to another. Depending on exactly which version of DOS you are using, you may have one or both of these commands available to you.

The listing for VC.BAT automates the procedure. Simply use Edlin, Edit, or a word processor that lets you save ASCII files to type in the listing and save it as VC.BAT.

As written, the program uses the FC command to compare the files. You may substitute the COMP command if FC isn't available on your system. Employing the program is easy. You may, for example, copy a file from drive C to drive A and verify it by typing the following at the DOS prompt:

```
VC MYFILE.EXT A:
```

This version of VC.BAT verifies the contents of files you copy with the DOS COPY command. As written, the program requires FC.EXE, but you may use COMP.EXE instead.

```
@ECHO OFF
IF ""=="%1" GOTO HELP
IF ""=="%2" GOTO HELP
IF NOT EXIST %1 GOTO NOSOURCE
COPY %1 %2
MD \VC.$$$
IF NOT "%2"==" " GOTO COMPARE
ECHO\> \VC.$$$\INPUT.$$$
:COMPARE
ECHO N>> \VC.$$$\INPUT.$$$
FC %1 %2 < \VC.$$$\INPUT.$$$
DEL \VC.$$$\INPUT.$$$
RD \VC.$$$
GOTO END
:HELP
ECHO VC copies files and verifies the source file
ECHO against the copy. To use VC, you must supply a
ECHO source and destination as you would with the
ECHO DOS COPY command.
GOTO END
:NOSOURCE
ECHO Source file not found. No files copied.
:END
```

End

Or you may copy to drive A and verify all files in the current directory on drive C by typing the following:

```
VC *.* A:?
```

POWER DELETING

Do you ever get tired of using the DEL command repeatedly to rid your hard drive of unwanted files? Although many file managers allow you to tag a number of files and delete them simultaneously, I sometimes find it more convenient to do my deleting directly at the DOS prompt—particularly if I first issue a DIR command so that I can see a list of the files I'll be removing. The batch program DELETE.BAT lets me do just that.

Simply type DELETE and then list the name of each file you want to erase, leaving a space between each filename, as in this command:

```
DELETE ONEFILE.TXT ANOTHER.FIL ATHIRD.ASC
```

(The command that you create may not exceed 127 characters.)

When the program executes, it first checks for the ex-

DELETE.BAT makes it easy to delete several files while working at the DOS prompt.

```
@ECHO OFF
REM DELETE.BAT
REM S. L. CROSSON
```

```
IF (%1)==() GOTO NOPARM
```

```
ECHO Ready to delete files entered.
ECHO Press Ctrl-Break at any time
ECHO to abort, or
PAUSE
```

```
:AGAIN
ECHO Deleting %1
DEL %1
SHIFT
IF NOT (%1)==() GOTO AGAIN
GOTO END
```

```
:NOPARM
ECHO You must provide the name(s)
ECHO of the file(s) to delete.
:END
```

End

istence of a command-line parameter (a filename), directs you to press Ctrl-Break at any time if you want to halt the program, and then deletes the specified files one by one.

This little batch program is a big time-saver when you need to delete several files from a directory and their filenames are so different that it's impossible to use wildcards to delete them.

—Stephen L. Crosson

CLEAN YOUR DISK WITH QBasic

Do your applications clutter your hard disk with numerous unwanted backup files? Many popular software packages, such as Microsoft Word, routinely create a backup file with the extension BAK every time you save a document.

Needless to say, it makes sense to delete these files periodically in order to free up valuable hard disk storage. The short QBasic program, DELBAK.BAS, deletes all the BAK files from your hard drive, except for hidden, system, and read-only files.

You could write a batch program to perform the same task, but DELBAK.BAS is much simpler, containing just 12 program lines. DELBAK.BAS works by using QBasic's SHELL command to execute DOS commands from within QBasic. It makes use of the DIR command's /S and /B switches, which were introduced in DOS 5. (QBasic is the programming language included with MS-DOS 5 and later.)

To enter the program listing below, type QBasic at the DOS prompt. When you're in QBasic, type each line of the listing as shown. The comments following the single quote on each line are optional; you may omit them if you wish. Check your typing, then select Save from the File menu. When prompted for the filename, type DELBAK.BAS. To run the program, select Start from the Run menu or press Shift-F5.

If you change the third line of the program, you may use it to delete files with extensions other than BAK. For instance, if you want to delete all files with the extension TMP, change the third line to read:

```
SHELL "DIR/S/B *.TMP > DUMPSTER"
```

Unlike DOS's DEL command, DELBAK.BAS deletes all BAK files from your hard drive without warning. So, before running the program, be sure you want all your BAK files eliminated.

—Mike DeAndrea

DELBAK.BAS deletes unwanted files from your hard drive using QBasic.

```
PRINT "Locating all BAK files." 'Print screen message.
SHELL "CD \" 'Change to root directory.
SHELL "DIR/S/B *.BAK > DUMPSTER" 'Find all BAK files, suppress
                                'directory header, and redirect
                                'output to dumpster file.
OPEN "DUMPSTER" FOR INPUT AS #1 'Then open dumpster file.
DO UNTIL EOF(1)
  LINE INPUT #1, BAKFILE$ 'Read in one filename at a time.
  PRINT "Deleting " + BAKFILE$ 'Print name of file being deleted.
  SHELL "DEL " + BAKFILE$ 'Delete file.
  LOOP 'Loop back to get next filename.
CLOSE 'Close dumpster file.
SHELL "DEL C:\DUMPSTER" 'Delete dumpster file.
END
```

End

text file that contain two text strings or look for lines that contain one text string but not another. To overcome these limitations, I wrote FINDIT.BAT, a batch program that adds the logical operators AND, OR, and NOT to the FIND command.

To use the program, type FINDIT at the DOS prompt, following the batch program's name with the name of the file to be searched and a text string for which to search. You must separate multiple strings by at least one logical operator. When you use AND, a line is displayed only if both of the text strings you specify appear in

the line. With OR, a line is displayed if it contains either of the specified strings. When you use NOT, a line is displayed only if it doesn't contain a particular text string.

For example, to search a file called MICHELLE.DOC for the strings *Paul* and *George*, type:

```
FINDIT MICHELLE.DOC Paul AND George
```

FINDIT.BAT then displays on screen any lines in MICHELLE.DOC that contain both *Paul* and *George*. Note that FINDIT.BAT, unlike the FIND command, doesn't expect you to place quotation marks before and after each string for which you want to search. Because of this,

MAKING FIND LOGICAL

DOS's FIND command lets you search through a text file for a specific string of text and list the lines in the file containing that string. Using the command's switches, you may display only files that don't contain the text string, place the file's line number in front of the line displayed, and—if you have DOS 5 and later—tell FIND to ignore differences in upper- and lowercase letters during its search.

Unfortunately, FIND doesn't let you do logical searches. That is, you can't use FIND to search for all lines in a

FINDIT.BAT overcomes many of the FIND command's limitations, adding the ability to use the logical operators AND, OR, and NOT. The program also lets users who have DOS 5 and later conduct searches that aren't sensitive to the case of the characters used in a search string.

```
@ECHO OFF
IF "%1"==" " GOTO DONE
IF "%1"=="I" GOTO SWITCH
IF "%1"=="i" GOTO SWITCH
SET SWITCH=
GOTO CONTINUE
:SWITCH
SET SWITCH=/I
SHIFT
:CONTINUE
SET FILESPEC=%1
SHIFT
FIND %SWITCH% "%1" %FILESPEC% >FOUNDIT.$$$
:RESUME
SHIFT
SHIFT
IF "%0"=="and" GOTO AND
IF "%0"=="AND" GOTO AND
IF "%0"=="or" GOTO OR
IF "%0"=="OR" GOTO OR
IF "%0"=="not" GOTO NOT
IF "%0"=="NOT" GOTO NOT
```

```
TYPE FOUNDIT.$$$ | FIND /V '-----' | MORE
GOTO DONE
:AND
FIND %SWITCH% "%1" FOUNDIT.$$$ >TEMP.$$$
DEL FOUNDIT.$$$
REN TEMP.$$$ FOUNDIT.$$$
GOTO RESUME
:OR
FIND %SWITCH% "%1" %FILESPEC% >TEMP.$$$
COPY FOUNDIT.$$$ + TEMP.$$$ FOUNDIT.$$$ >NUL
DEL TEMP.$$$
GOTO RESUME
:NOT
FIND /V %SWITCH% "%1" FOUNDIT.$$$ >TEMP.$$$
DEL FOUNDIT.$$$
REN TEMP.$$$ FOUNDIT.$$$
GOTO RESUME
:DONE
SET SWITCH=
SET FILESPEC=
DEL FOUNDIT.$$$
```

End

FINDIT.BAT doesn't let you search for text strings that contain a space.

When using FINDIT.BAT, remember that DOS processes command-line parameters from left to right. This becomes important in commands such as the following:

```
FINDIT APPLE.DOC John AND Paul NOT George OR Ringo
```

When you first look at this line, it seems that it instructs FINDIT.BAT to list all the lines that contain *John* and *Paul* but not *George* or *Ringo*. But that's not the case. Because DOS processes the parameters from left to right, the command first finds all lines that contain *John* and *Paul*. Next, if any lines contain *George*, they are excluded. Finally, all remaining lines are displayed, because OR indicates that it doesn't matter whether or not they contain *Ringo*.

To rewrite the command so that it finds all lines that contain both *John* and *Paul* but don't contain either *George* or *Ringo*, use the following:

```
FINDIT APPLE.DOC John AND Paul NOT George NOT Ringo
```

To make the search insensitive to the case of the characters in a string, simply place an *I* on the command line prior to entering the filename, as in:

```
FINDIT I MICHELLE.DOC Paul AND George
```

If FINDIT.BAT doesn't locate any matches when it looks for a search string, it prints a blank line. If you happen to run out of environment space while running FINDIT.BAT, you may increase its size from DOS.

Although FINDIT.BAT has limitations, it makes the DOS FIND command more powerful, providing logical search features even most word processors and commercial file-search utilities lack.

—Vincent D. O'Connor

FILTERING OUT ENTER

If you're a batch-file programmer, you've probably used the ECHO command to create other files. But when you do this, ECHO always adds the equivalent of the Enter key after the echoed information. Subsequent data written to the file appears on a second line, which is frustrating if you need to put it at the end of the first line.

Fortunately, you can sidestep this problem by changing the ASCII characters that represent the Enter key—a carriage return (ASCII 13) followed by a line feed (ASCII 10)—into spaces. The easiest way to make

minor changes of this sort to a file is to create a filter. You send the file to the filter as input and redirect the output of the filter to a new file. The new file contains the changes the filter made. The best tool for making a simple filter is Debug.

The filter I developed does a little more than simply remove carriage returns and line feeds, however. NOCTRL.COM changes all control codes, or all characters with ASCII values of less than 32, into spaces. It reads its input one character at a time from standard input and checks each character. If it finds a control character, it sends a space to standard output. If the input character isn't a control character, NOCTRL.COM simply sends it to standard output. The filter is done when it comes to the end of its input.

To create NOCTRL.COM, use a text editor to type in the listing (be sure to include the blank line) and name it NOCTRL.SCR. After making sure that Debug is available in the current directory or in your PATH statement, issue this command at the DOS prompt:

```
DEBUG < NOCTRL.SCR
```

Debug then creates NOCTRL.COM, storing it in the current directory.

COMBINING FILES

Periodically, I print out a copy of all the pieces of a work in progress—from modules for a programming project to fragments of research for an article. I quickly got tired of handling this tedious task manually and decided to create a batch program that would take care of things. Along the way, it struck me that it also would be useful to gather selected parts of a set of files. The re-

NOCTRL.SCR is a Debug script that lets you create NOCTRL.COM, which removes all carriage returns and line feeds from a file by changing all control codes into spaces.

```
N NOCTRL.COM
A 100
MOV AH,3F
SUB BX,BX
MOV CX,1
MOV DX,80
INT 21
JC 0125
CMP AX,1
JL 0125
MOV SI,80
MOV DL,[SI]
CMP DL,20
JNB 011F
MOV DL,20
MOV AH,2
INT 21
JMP 0100
INT 20
RCX
27
W
Q
```

End

COMB.BAT lets you combine a set of files or compile selected references from a set of files.

```
@ECHO OFF
REM COMB.BAT
REM This program sends a series of files to the
REM screen, a printer, a communications port, or a
REM disk file. If you wish, you may use the program
REM to search a series of files and send only
REM fragments of them. Its many uses include
REM combining a set of files into a single archive
REM file; searching through a group of files and
REM printing all references to a particular date,
REM name, or other reference; or creating a document
REM log. Although the program assumes it's working
REM with text files, it also may work with nontext
REM files. However, the results are difficult to
REM predict. The program must be able to find the
REM file CR.FIL, which contains a single, blank line.
REM As written, the program looks for CR.FIL in the
REM root directory.

IF "%1" == "?" GOTO SYNTAX
IF "%1" == "" GOTO MISSING
IF "%2" == "" GOTO MISSING

SET TARGET=%3
IF "%3"=="" SET TARGET=CON
DATE < \CR.FIL | FIND "is" >> %TARGET%
TIME < \CR.FIL | FIND "is" >> %TARGET%
ECHO. >> %TARGET%

IF "%2" == "$$" GOTO NO_SEARCH

ECHO Searching for %2 >> %TARGET%
ECHO.
```

```
FOR %%F IN (%1) DO FIND "%2" %%F >> %TARGET%
GOTO END

:NO_SEARCH
FOR %%F IN (%1) DO TYPE %%F >> %TARGET%
GOTO END

:MISSING
ECHO.
ECHO Parameter missing.

:SYNTAX
ECHO.
ECHO Syntax: COMB [source_file_spec]
ECHO [search_string/$$] [destination]
ECHO.
ECHO [source_file_spec], the file or files (if you
ECHO use a wildcard) you want to search through;
ECHO may include a path.
ECHO [search_string], the group of characters to
ECHO search for in the source file(s). The program
ECHO displays only those lines of the source
ECHO file(s) that contain the search string.
ECHO To prevent a search, substitute $$ for the
ECHO search string. The program types entire file(s).
ECHO [destination] is optional. You may substitute
ECHO the name of a disk file; or the printer
ECHO designation, PRN; or a COM port. If this
ECHO parameter is omitted, the program sends the
ECHO files to your monitor.
:END
```

End

sult of my efforts is COMB.BAT, a batch file that lets you combine a set of files or compile selected references from a set of files.

COMB.BAT requires one helper file named CR.FIL, which you may create using the COPY command. At the DOS prompt, type:

```
COPY CON \CR.FIL
```

and press Enter. On the next line, press Enter again. On the following line, terminate the file by typing Ctrl-Z or pressing the F6 key and pressing Enter. This creates a two-byte file containing a single carriage return.

If you run COMB.BAT without supplying any parameters on the command line, it displays its command syntax and a short explanation of how to use the program. It then checks the search term you supplied as the second command-line parameter to see if it is the unique signal (\$\$) that indicates you want to process entire files.

The next two sections of the program have separate functions. The first section lets you use the FIND com-

mand to pick out fragments of files that meet your search criteria; the second sends entire files to your target device or file.

COMB.BAT can combine files under a number of useful circumstances, doing the hard work so that you don't have to.

—Harry Bee

MAKING UNIQUE FILENAMES

If you ever have copied a file and inadvertently written over another file of the same name, then you know how important it is to assign unique names to your files. My QBasic program, TIMENAME.BAS, helps you do just that by using the time and date to assign filenames. TIMENAME.BAS works by creating a batch program called SETNAME.BAT, which consists of only two lines. SETNAME.BAT sets an environment variable

TIMENAME.BAS is a QBasic program that uses the current date and time to assign unique names to files.

```
*****TIMENAME*****
'
'To run the program from the command line or
'within a batch program, type:
'
'    QBASIC /RUN TIMENAME.BAS
'
'Your \BAS directory must be listed in your PATH
'statement.
'
*****
'Create three "lookup" strings:
MONTH$ = "ABCDEFGHJKLMN"
DAY$ = "1234567890ABCDEFGHJKLMNPQRSTUVWXYZ"
HOUR$ = "1234567890ABCDEFGHJKLMNPQ"

'Get the current date and time:
STR1$ = DATE$
STR2$ = TIME$

'Generate index numbers from the date and time:
MONTHNUM = VAL(MID$(STR1$, 1, 2))
DAYNUM = VAL(MID$(STR1$, 4, 2))
HOURNUM = VAL(MID$(STR2$, 1, 2))
MINNUM = VAL(MID$(STR2$, 4, 2))
```

```
SECNUM = VAL(MID$(STR2$, 7, 2))
```

'Create the string TIMENAME\$ by concatenating DATE\$,
'the "lookup" results, and HEX\$:

```
TIMENAM1$ = MID$(STR1$, 10, 1) + MID$(MONTH$,  
    MONTHNUM, 1)
```

```
TIMENAM2$ = MID$(DAY$, DAYNUM, 1) + MID$(HOUR$,  
    HOURNUM, 1)
```

```
TIMENAM3$ = HEX$(MINNUM) + HEX$(SECNUM)
```

```
TIMENAME$ = TIMENAM1$ + TIMENAM2$ + TIMENAM3$
```

'Print it to the screen:

```
PRINT TIMENAME$
```

'Open the output file and write two lines to it:

```
OPEN "SETNAME.BAT" FOR OUTPUT AS #1
```

```
PRINT #1, "@ECHO OFF"
```

```
PRINT #1, "SET TIMENAME=" + TIMENAME$
```

'Close the output file and return control to the

'system:

```
CLOSE #1
```

```
SYSTEM
```

End

called TIMENAME, which then is used to generate the filenames.

The batch program shown below, CFGBAK.BAT, shows how you might use TIMENAME.BAS to save backup copies of files. It creates backup versions of your AUTOEXEC.BAT and CONFIG.SYS files, storing them in a directory called C:\CFGBAK. With TIMENAME.BAS and CFGBAK.BAT, it's easy for me to maintain multiple copies of my start-up files.

—Douglas Park

CFGBAK.BAT is an example batch program that shows how you might put TIMENAME.BAS to work creating backup files and assigning them unique filenames.

```
REM Run QBasic from the command line.
REM TIMENAME.BAS will return you to DOS.
QBASIC /RUN TIMENAME.BAS
```

```
REM Call the batch program, then delete it:
```

```
CALL SETNAME.BAT
```

```
DEL SETNAME.BAT
```

```
REM Use the environment variable to make filenames:
```

```
COPY C:\AUTOEXEC.BAT C:\%TIMENAME%.BAT
```

```
COPY C:\CONFIG.SYS C:\%TIMENAME%.SYS
```

```
REM Remove the environment variable:
```

```
SET TIMENAME=
```

End

A POOR MAN'S DATA COMPRESSION

After a careful study of my computing habits, I was shocked to discover that I use 30 percent of my programs only once per month and another 20 percent only once or twice a year. Because these programs take a lot of hard disk space, my first thought was to remove them. However, the amount of work required to reinstall these files from their distribution disks was a convincing argument for keeping them permanently stored on my hard drive.

Then I discovered PKZIP.EXE, PKWARE's great file-compression utility. Beginning with version 2.04g, it lets you archive an entire branch of a DOS subdirectory tree in a single ZIP file; its companion, PKUNZIP.EXE, decompresses the files and restores the underlying directory structure.

I decided to make the two programs the heart of a batch file that compresses my seldom-used programs and their associated data files and decompresses them when I need them. The result of my labors, JAMMIT.BAT, not only decompresses the files in a directory, but it also can start a compressed application and put compressed files back into the archive when the job is done. JAMMIT.BAT shows you how much disk space you need to decompress a ZIP directory and how much hard disk space is available.

Although JAMMIT.BAT doesn't save much space if you

JAMMIT.BAT uses the file-compression program PKZIP to compress and decompress directory branches on the fly.

```
@ECHO OFF
REM -----
REM JAMMIT.BAT uses PKZIP.EXE and PKUNZIP.EXE 2.04g
REM to reduce hard disk clutter and conserve disk
REM space by archiving (compressing) all files in a
REM branch of your DOS directory tree. PKZIP.EXE
REM and PKUNZIP.EXE are copyrighted shareware
REM programs: PKWARE Inc., 9025 N. Deerwood Dr.,
REM Brown Deer, WI 53223
REM -----
REM
IF "%1" == "" GOTO HELP
IF "%2" == "" GOTO HELP
IF "%3" == "" GOTO HELP
CD %1
IF EXIST %1%3 GOTO NOZIP
ECHO.
ECHO Performing initial compression of %1%2 to %3
PKZIP %1%3 -ex -m -rP -whs %1%2\*.*
ATTRIB +r %1%3
:NOZIP
PKUNZIP %1%3 -vb
DIR %1 > FREE.DAT
FIND "bytes free" %1free.dat
ECHO.
ECHO The first number (centered above) indicates free
ECHO space available on %1. The second number (far
ECHO left column above) is the disk space required by
ECHO your original (uncompressed) files, which are
ECHO now archived in %3.
ECHO.
ECHO Subtract the second number from the first
ECHO number.
ECHO.
ECHO If there is insufficient free disk space on %1
ECHO to decompress the files in %3, you should press
ECHO Ctrl-C now to abort.
ECHO.
ECHO Press any other key to decompress the files from
ECHO %3 to %1 and start your application program %4.
ECHO.
DEL %1free.dat
PAUSE
ECHO.
ECHO Decompressing files for %1%2.
ATTRIB -r %1%3
PKUNZIP -d -Jrhs %1%3
DEL %1%3
CD %1%2
IF "%4" == "" GOTO RESTORE
IF NOT EXIST %1%2\%4 GOTO NOAPP1
```

```
ECHO.
ECHO Starting application program: %1%2\%4
%4
GOTO RESTORE
:NOAPP1
ECHO Cannot start %4. Application not found in
ECHO archive %3.
:RESTORE
ECHO.
ECHO Press Ctrl-C, then press Y to restore files in
ECHO %1%2 to their original (decompressed) state,
ECHO delete archive file %3 from %1, and return to
ECHO the DOS prompt.
ECHO.
ECHO Press any other key to rearchive the files in
ECHO %1%2 and return to the DOS prompt.
ECHO.
PAUSE
CD %1
PKZIP %1%3 -ex -m -rP -whs %1%2\*.*
ATTRIB +r %1%3
ECHO.
ECHO Compression of files in %1%2 completed.
GOTO ENDIT
:HELP
ECHO.
ECHO SYNTAX: JAMMIT drive:\ dirname zipname.zip
ECHO appname.ext
ECHO.
ECHO drive is a valid drive letter, followed by :\.
ECHO.
ECHO dirname is a valid directory name on drive.
ECHO.
ECHO zipname.zip is the archive in which you want
ECHO to store the compressed files. The file
ECHO extension ZIP is required.
ECHO.
ECHO appname.ext is the name of the application
ECHO program you want to start after the archive
ECHO has been decompressed. It must include a COM,
ECHO EXE, or BAT file extension.
ECHO.
ECHO Note: This parameter is optional.
ECHO You may leave it blank if you do not wish
ECHO to start an application, or for archiving
ECHO directories that contain only data files.
ECHO.
ECHO Example: JAMMIT C:\ QPRO QPRO.ZIP Q.EXE
ECHO.
:ENDIT
```

End

use it on a system running disk-compression software, such as DOS 6.x's DoubleSpace or Stac Electronics' Stacker, you still may find it useful. Because the batch program stores an entire branch of your directory tree in one ZIP file, it can help organize an unruly hard disk.

Use DOS's Edit text editor to type in the listing and name the file JAMMIT.BAT. Be sure to type the commands exactly as shown. Although the syntax used in some lines—sequences such as %1%3, for instance—may look crunched together, it is correct as written. Don't add extraneous spaces. Next, make sure that

JAMMIT.BAT, PKZIP.EXE, and PKUNZIP.EXE are stored in a directory listed in your PATH statement.

Unzip That File

JAMMIT.BAT uses the following syntax to perform both compression and decompression:

```
JAMMIT drive:\ dirname zipname.ZIP [appname.ext]
```

drive: specifies the root directory of the drive where your application and its data files are stored. *dirname* names the directory where your application and data files are stored. If your application resides two or more levels below the root directory (if, for example, the file is in *dirname1\dirname2*), use a backslash separator between the names. Leading or trailing backslashes aren't permissible, however.

ZIPNAME specifies the name under which you want to store your archive file. The file extension ZIP is required. APPNAME specifies the name of the application that you want to start after the files are decompressed; you must add a COM, EXE, or BAT extension. If you are compressing only data files, you may omit this parameter.

JAMMIT.BAT assumes that your application resides in its own subdirectory one level below the root directory. PKZIP automatically compresses the files in the directory, including all subdirectories that lie below it in the DOS directory tree.

For example, one of the applications I seldom use is Borland's Quattro Pro spreadsheet. To archive this application, I type:

```
JAMMIT C:\ QPRO QPRO.ZIP Q.EXE
```

Pay special attention to the blank space between the drive parameter and the directory parameter. Although it may seem logical to join C:\ and QPRO in the above example, the blank space is required, because the drive and directory are separate, replaceable parameters used by the batch program.

If you name an application when you invoke JAMMIT.BAT, the batch program then starts the application. After you complete your work, exit your application as you usually do; the batch program resumes control automatically.

The program offers you the chance to restore original files and erase the compressed directory. If you press Ctrl-C and type Y, the application and all other files in its directory branch are restored to their decompressed state. If you press keys other than Ctrl-C, PKZIP recompresses the files into the archive file.

—John Wolfskill

DIRCOMP.BAT is a batch program that lets you compare a disk or disk directory against your backup files.

```
@ECHO OFF
REM Directory Comparison Batch program
ECHO *****
ECHO * This program creates two files in the *
ECHO * current directory. They are named *
ECHO * CURRENT and OTHER. *
ECHO *****
PAUSE
ECHO ON
DIR %1/%2 | SORT > OTHER
DIR %2 | SORT > CURRENT
FC OTHER CURRENT | MORE
```

End

COMPARING DIRECTORIES

Here's a batch program that will perform a quick comparison between a disk or disk directory and your backup files. The program, called DIR.COMP, uses three DOS utilities—SORT.EXE, MORE.COM, and FC.EXE; they must be available in the same directory or search path as the batch program.

DIRCOMP.BAT works well in a situation in which a team of programmers are working together in order to develop a single application. All team members can keep copies of the source and data files on their hard disks.

To compare two directories, simply type the command:

```
DIRCOMP directory filespec
```

where *directory* is the directory that you wish to compare with the contents of the currently logged directory, and *filespec* is the name of the file that you want to compare. The file specification may be the name of an individual file or contain the DOS wildcard characters ? and *.

After DIRCOMP creates the output files—named CURRENT and OTHER—it compares the two until it finds a line that is not the same in both files. It then displays the name of the first file and the lines that differ, followed by the name of the second file and the corresponding lines from that file. The important differences to note are the time and date stamps.

If differences exist in three or fewer lines in both files, a lengthy report is issued. If there are numerous differences, however, DOS's FC program reports "Files are different." If your version of DOS doesn't include FC.EXE, use COMP.EXE, instead.

HARDWARE

HELPERS

Batch files solve many computing problems, but sometimes only a *real* program can make your PC do what you want done. Fortunately, you don't need a lot of expensive tools for such jobs. Armed with Edit, QBasic, and Debug, you've got everything you need to write programs in Basic and assembly language.

Need to swap a printer port, execute certain programs on certain days, or control the sometimes troublesome Num Lock key? Look no further. The following 12 programs can solve these problems—and more.

PUT MORE LINES ON YOUR SCREEN

If you'd like to see more than the typical 25 lines of text on your screen, perhaps DOS can help. Many monitors and video adapters let you use the DOS MODE command to switch to a 43- or 50-line display. To find out whether your video components permit this, first be sure your CONFIG.SYS file contains a statement that loads the ANSI.SYS device driver. On most systems, ANSI.SYS resides in the C:\DOS directory, so the correct command is:

```
DEVICE=C:\DOS\ANSI.SYS
```

If ANSI.SYS is stored in another directory, substitute the correct path.

After adding a command that loads ANSI.SYS and saving the new version of CONFIG.SYS, restart your computer to have the new command take effect. Then, at the DOS prompt, type the following:

```
MODE CON: LINES=43
```

Next, issue a command such as DIR to see how things look when DOS puts 43 lines on screen. Then take a look at a 50-line display by typing:

```
MODE CON: LINES=50
```

Issue another DIR command. If you like either the 43- or 50-line display well enough to look at it day in and day out, add the appropriate MODE command to your AUTOEXEC.BAT file.

Not all display adapters support the 43- and 50-line settings, so be sure to try the commands from the DOS prompt before adding one of them to your AUTOEXEC.BAT file.

—Jeff DeTray

A BASIC SCREEN SAVER

Screen saver programs are all the rage these days. They allow you to walk away from your system for long periods of time without turning off the monitor to prevent a static screen image from “burning into” the display. If this sounds like something you think that you need, perhaps you should try the program SAVER.BAS. It's not fancy—you won't find any flying toast-ers—but it does the job and requires minimal disk space. The program works equally well in GW-Basic and QBasic.

To create and use SAVER.BAS, start your version of Basic, type in the listing, save it as SAVER.BAS, and run it. You'll see a yellow line, which “bounces” around the screen until you press a key. Once SAVER.BAS is working properly, create a short batch file called SAVE.BAT to make it easier to activate the program from the DOS prompt.

If you're running GW-Basic and both GWBASIC.EXE and SAVER.BAS are in your C:\DOS directory, SAVE.BAT will look like this:

```
@ECHO OFF
CD\DOS
GWBASIC SAVER
```

The simple screen saver SAVER.BAS displays a yellow line that bounces around your display.

```
100 'Moving line screen saver using GW-Basic
    graphics
110 'D.Ferrier
120 '-----initialize line-----
130 SCREEN 9: CLS: COL = 14
140 X1DIFF = -2 : Y1DIFF = 5: X2DIFF = -3: Y2DIFF
    = 4
150 X1=50: Y1=100: X2=200: Y2=300
160 '-----initialize line-----
170 WHILE 1=1
180 X1 = X1+X1DIFF: Y1 = Y1+Y1DIFF
190 X2 = X2+X2DIFF: Y2 = Y2+Y2DIFF
200 LINE (X1V,Y1V)-(X2V,Y2V), 0
210 LINE (X1,Y1)-(X2,Y2), COL
220 X1V = X1: Y1V = Y1: X2V = X2: Y2V = Y2
230 IF X1 > 640 OR X1 < 1 THEN X1DIFF = X1DIFF*-1
240 IF Y1 > 350 OR Y1 < 1 THEN Y1DIFF = Y1DIFF*-1
250 IF X2 > 640 OR X2 < 1 THEN Y2DIFF = Y2DIFF*-1
260 IF Y2 > 350 OR Y2 < 1 THEN Y2DIFF = Y2DIFF*-1
270 IF INKEY$ <> "" THEN SYSTEM
280 WEND
```

End

A version of SAVE.BAT for QBasic users will look like this:

```
@ECHO OFF
CD\DOS
QBASIC /RUN SAVER
```

(Be sure to store SAVE.BAT in a directory listed in your PATH statement.) Then, whenever you are at the DOS prompt and want to activate the bouncing-line display, simply type SAVE and press the Enter key.

—David Ferrier

COLORFUL START-UPS

Tired of DOS's boring old text white-on-black screen display? Why not try STARTCOL.COM, a program that lets you change the settings of DOS's foreground and background colors. You may place STARTCOL.COM in your AUTOEXEC.BAT file to set screen colors at start-up or change screen colors on the fly by typing STARTCOL at the DOS prompt.

The program uses your computer's ROM BIOS video-handling routines to change the color settings. When you run an application, it displays the text characters in the foreground color you select. You may revert to DOS's standard display by typing CLS.

STARTCOL.SCR requires that you express your choice of

SCREEN COLORS

Background (0-7)		Foreground (0-F)	
Black	0	Black	0
Blue	1	Blue	1
Green	2	Green	2
Cyan	3	Cyan	3
Red	4	Red	4
Magenta	5	Magenta	5
Yellow	6	Brown	6
Gray	7	Lt. Gray	7
		Gray	8
		Lt. Blue	9
		Lt. Green	A
		Lt. Cyan	B
		Lt. Red	C
		Lt. Magenta	D
		Yellow	E
		White	F

foreground and background colors as a hexadecimal (hex) number between zero and 7F. Use the table "Screen Colors" to select the number or letter that corresponds to the background and foreground colors you like best. Then replace the question marks in the eleventh line of the Debug script STARTCOL.SCR with the two-digit number representing those colors. If, for example, you select blue (1) as the background color and white (F) for the foreground, you must put the two numbers side by side to create the hex number 1F and substitute this value for the question marks in MOV BH,??.

First, remove ANSI.SYS from CONFIG.SYS. Then create STARTCOL.SCR using Edit or a word processor that produces ASCII (text) files. Don't type the semicolons or the comments that appear to the right of them. Next, create the program from this script by typing:

```
DEBUG < STARTCOL.SCR
```

If you like more than one color combination, you may make several versions of the program, but be sure to give each copy of the program a unique name.

STARTCOL.SCR, a Debug script that creates the executable program STARTCOL.COM, lets you set screen colors at start-up or on the fly.

```
N STARTCOL.COM
A 100
MOV AH,2 ;Move cursor
MOV BH,0 ;Page 0
XOR DX,DX ;Row 0, col 0
INT 10 ;Video BIOS
MOV AX,600 ;Scroll screen
MOV BH,?? ;Insert color here
SUB CX,CX ;Entire screen
MOV DX,184F ;Columns,rows
INT 10 ;Video BIOS
RET

RCX
15
W
Q
```

End

CONTROLLING YOUR CURSOR

Although ANSI.SYS provides commands that let you locate the cursor on screen, it doesn't offer commands that turn the cursor on and off. The Debug scripts CUROFF.SCR and CURON.SCR create two short programs that compensate for this. Using a text editor that saves files in ASCII format, type in the listings. Convert the scripts to programs by typing the following commands at the DOS prompt:

```
DEBUG < CURON.SCR
DEBUG < CUROFF.SCR
```


By exclusive arrangement with PowerBASIC,
DOS Resource Guide brings you

POWER BASIC 2.1

and Don Inman's

"Learning Basic"

(a book written specifically for the PowerBASIC user)—at the combined low price of

For anyone who has ever written or typed in a BASIC program, this is a remarkable opportunity. Here's a chance to enrich your BASIC programming skills and acquire a powerful programming language at a fraction of the regular cost.

only ~~\$129.95~~
\$39.95

12 free*
BONUS PROGRAMS!

*ORDER by April 30th and we'll include 12 BONUS BASIC programs from DOS Resource Guide ABSOLUTELY FREE!

OK YES!

Here's what you'll get:

PowerBASIC version 2.1—a complete environment for writing, running, debugging, and compiling BASIC programs that includes:

Compiler—create your own stand-alone executable programs.

Powerful Language Features—faster and with more features than any other BASIC.

Integrated Development Environment—lets you write, edit, run, and debug programs quickly and easily.

Great Debugging Tools—helps you find and correct elusive program bugs in seconds.

Help System—extensive context-sensitive help always available to make learning to program easy and fun.

Compatibility—most programs written in other BASICs require only minor modifications.

"Learning BASIC"—a step-by-step tutorial on the ins and outs of PowerBASIC. Written by an experienced BASIC instructor, this book will help you:

- Become familiar with the PowerBASIC programming system
- Learn the BASIC language one step at a time
- Write and edit your own programs, functions and subroutines
- Discover techniques for effective troubleshooting
- Go beyond the basics to advance your programming skills

.....and much more!

This is a limited-time offer
so call or mail to reserve your copy today!

Please send me:

**POWER BASIC 2.1
& "Learning Basic"**

No. of Orders () at \$39.95* each

Name _____

Address _____

City _____ State _____ Zip _____

☐ Payment enclosed

Charge my: ☐ VISA ☐ MC ☐ AmEx ☐ Discover

Card# _____

Expiration Date _____

Signature _____

*****30-DAY MONEY-BACK GUARANTEE !**

If after 30 days, you are not completely satisfied with your purchase, just return the merchandise with a note and we will gladly refund every penny!

Please allow 4-6 weeks delivery. For faster service, please inquire. Prepayment is required on all orders. U.S. funds on U.S. banks only. *Canada, Mexico, and foreign surface, please add \$5.00 per item ordered. Airmail, please add \$9.50.

Mail To: DOS Resource Guide
P.O. Box 672, Mt. Morris, IL 61054

AS4PB

**DON'T MISS THIS UNIQUE OPPORTUNITY!
CALL AND RESERVE YOUR COPY TODAY:**

1-800-349-7327

CUROFF.SCR is a Debug script that creates **CUROFF.COM**, a program that turns off DOS's cursor.

```
N CUROFF.COM
A 100
MOV AH,3
INT 10
OR CH,20
MOV AH,1
INT 10
RET

RCX
C
W
Q
```

End

CURON.SCR is a Debug script that creates **CURON.COM**, a program that turns on DOS's cursor.

```
N CURON.COM
A 100
MOV AH,3
INT 10
AND CH,DF
MOV AH,1
INT 10
RET

RCX
C
W
Q
```

End

These commands create **CURON.COM** and **CUROFF.COM**, which let you turn your cursor on and off. You may execute these programs at the DOS prompt or invoke them in your batch programs. (Don't worry about adding a **DEVICE=ANSI.SYS** command to your **CONFIG.SYS** because program relies on **ANSI.SYS**.)

—Jane Grisetti

I BRAKE FOR GAMES

If you own game software that predates today's faster 386 and 486 PCs, you may run into a program that executes graphics so fast that the game is unplayable. Don't immediately assume that you must throw it out. The problem often is caused by timing-delay loops that

Use WHOA.SCR to create WHOA.COM, a program that slows down some games that were written for an older PC and that run too fast on a newer system.

```
N WHOA.COM
A 100
JMP 112
DB 0,0,0,0
PUSH CX
MOV CX,03E8
LOOP 010A
POP CX
CS:
JMP FAR [102]
MOV AH,35
MOV AL,1C
INT 21
MOV [102],BX
```

```
MOV [104],ES
MOV AH,25
MOV DX,0106
MOV AL,1C
INT 21
MOV DX,0112
INT 27
```

```
RCX
2E
W
Q
```

End

depend on instructions from the central processing unit for their operation. My program **WHOA.COM**, which is created from the Debug script **WHOA.SCR**, lets you solve this difficulty by slowing down your processor to an acceptable speed. (Unfortunately, **WHOA.COM** doesn't work with programs that time operation by monitoring the BIOS's time-of-day count.)

Using **Edlin**, **Edit**, or a word processor that saves files in **ASCII** format, type in the listing and save it as **WHOA.SCR**. Then return to **DOS** and assemble the program by typing:

```
DEBUG < WHOA.SCR
```

To put the program to work, type **WHOA** at the **DOS** prompt. This installs the program in memory and slows down your processor. Next, run the offending software. If running **WHOA.COM** once doesn't slow down the processor enough, exit the program and load another copy of **WHOA.COM**. Each time you load another copy, its time delay is added to any delays already in force. To remove any installed copies of **WHOA**, you must reboot your PC.

If you have **The Norton Utilities**, you may use **Sysinfo**, a utility for monitoring computer speed, to assist in the fine-tuning. Continue to load additional copies of **WHOA.COM** (each copy requires approximately 300 bytes of **RAM**) until **Sysinfo** shows that your system is operating at a computing index value of 1.0—the approximate speed of an **IBM PC**. Count the total number of times you loaded **WHOA.COM** to achieve the desired speed, and multiply this figure by the delay count in the **CX** register. Insert this new value in the **CX** register to create a copy of **WHOA.COM** that sets the computer to the precise speed that you want for an application.

—John Wolfskill

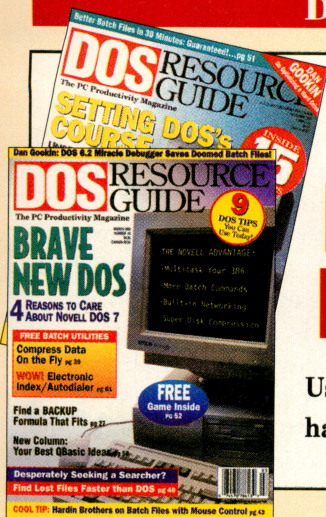
CHECK THE DATE

I wanted a program to run antivirus software and **CHKDSK** at regular intervals. I thought that writing a program in assembly language might be a fast solution, so I came up with **DATECHK.COM**, a program that updates a single file, **DATE.FIL**, whose date stamp is used to check when the last update was made. **DATECHK.COM** returns an error code of 1 if **CHKDSK** or another operation is needed and an error code of zero if all is well.

To be helpful, **DATECHK.COM** must be used in a batch file. Even when used in a batch file, however, the program executes quickly, because only one file is manipu-

THE COMPLETE DOS RESOURCE GUIDE

BACK ISSUES ORDER FORM



Make sure you
own all the best
**BACK
ISSUES!**

Use our toll-free number and
have your credit card ready!

DRG#5: The One-Stop DOS Primer

All you can eat: DOSKEY, SETVER, APPEND, and ASSIGN, plus a tutorial on using the ATTRIB command. Part 1 of a DEBUG series. Our "10 Worst" list of DOS commands. Your introduction to serial ports. FREE QBasic printer control program: just type and run. Sharpen your DOS finesse with 15 type-in tips.

DRG#6: 13 Ways to Boost Your DOS IQ

Peter Norton shares some valuable COPY secrets. Personalizing the DOS 5 shell. Part II: altering disk data with DEBUG. Modern communications made simple. Meet the DOS competition—our first look at DR DOS. Compression/decompression software explained. Part I of a batch-file clinic. FREE PROGRAM: a QBasic name and address database. Plus, 13 tips for boosting your DOS IQ.

DRG#7: Get Creative with DOS!

Instant programming with articles on DEBUG and batch files: You'll learn how to write your own DEBUG scripts and add error-code checking to a batch file. How to outline a QBasic program. Should you divvy up your new hard drive? Our picks for building a shareware library and advice on how to avoid 35 traps in a Windows setup. Plus, 15 tips to DOS happiness.

DRG#8: Getting CONFIG.SYS Right!

John Dvorak takes you step-by-step through the setup of CONFIG.SYS. Put personality in your PROMPT. A MIRROR command can save your data. Taking care of the environment in a batch file. FREE PROGRAM: A

perpetual calendar in QBasic. How to customize your modem. More on PATH, pens, and Windows Program Manager. Plus, our annual index to articles: DRG 1-7. And 8 tips add power to your command line.

DRG#9: Create Your Own DOS Game!

3 FREE games in QBasic: type and run. First of a series on ways to improve batch files using DEBUG. Create a database with the prompt command. Switch-hitting with DOS 5's DIR. Dan Cookin's new FOR-MAT.COM. 9 batch files from Ronny Richardson's collection that will move you from one subdirectory to another. 7 shareware utilities that make life easier at the DOS prompt. Plus 6 tips from the command line.

DRG#10: Are You Ready for DOS 6?

First look at MS-DOS 6: Double the disk storage, better backup. All about DELETE. Automate CHKDSK in a batch file. Memory management special: how DOS gets out of 640K, how memory utilities work faster and easier. Peter Norton gets your DOS applications started under Windows. FREE PROGRAM: Fraction Maker—decimal to fraction/fraction to decimal converter. How to beat hard-drive fatigue. 7 tips from the prompt.

DRG#11: Are DOS Utilities Doomed?

Utility publishers struggle successfully to outperform DOS 6 with its caching, memory, and hard disk add-ins. FREE: DEBUG script keeps checking for input. An easy batch file for killing a directory without DOS 6. Converting GW-Basic programs to QBasic. Q&A on

Super VGA. Using the new CHOICE command. FREE: QBasic loan calculator. 8 DOS tips for the command-line insecure.

DRG#12: Setting DOS's Course

The DOS environment statement is easier to understand than you think. All about FORMAT. A DEBUG script that launches a program after a countdown. Get started with batch files: Guaranteed! Disk caching explained. Using ERRORLEVELs in batch files. Cookin on DOS 6 DEFRAG. FREE: QBasic color finder. Plus 15 power tips from the prompt.

DRG#13: The DOS Wars Heat Up

IBM PC DOS vs MS-DOS. 10 ways to use DOSKEY. DoubleSpace changes how you use other commands. How to make batch files a little friendlier. Where to find DOS help on-line. Advanced class in QBasic—boost program speed with assembly-language assists. Index to DRG's DOS tips: Issues 1-12. Plus 13 new tips for the self-taught DOS pro.

DRG#14: Brave New DOS

4 reasons to care about Novell DOS 7. 2 FREE batch utilities: electronic Rolodex/autodialer and do-it-yourself data compression. Using MS-BACKUP. Redirecting input with a DEBUG script. Adding mouse control to a batch file. Debug batch files line by line. FREE PROGRAM: Hangman! Plus, 9 DOS tips for the C-prompt aficionado.

DRG#15: How to Beat Filename Frenzy

Advice on how to beat DOS's cryptic 8-character filenames convention. DoubleSpace produces an unexpected upside: flexible drive partitions. DEBUG script produces ASCII key values. New MOVEs from your old DOS. FREE Batch File: Scoot in and out of directories fast. Piping and redirection primer. PART II: More powerful QBasic—some Assembly required. Dan Cookin shows you 3 ways to set up a DOS program under Windows.

1-800-349-7327

BACK ISSUES of DOS Resource Guide are only \$5.95* postpaid!

AS4BA

Please enter below the quantity of each Back Issue needed.

DRG#5		DRG#9		DRG#13	
DRG#6		DRG#10		DRG#14	
DRG#7		DRG#11		DRG#15	
DRG#8		DRG#12			

I enclose \$5.95 for each issue. Please mail (post paid) to:

Name _____

Street _____

City _____

State/Zip _____

☐ Payment enclosed ☐ Check or MO

Charge my: ☐ VISA ☐ MasterCard ☐ AmEx ☐ Discover

Card # _____

Exp. Date _____

Signature _____

*Back issues to Canada and Mexico cost \$6.95 (postpaid); other international (surface) \$8.00, (air) \$13.00. Checks and money orders must be payable in U.S. funds, drawn on a U.S. bank. VISA, MasterCard, Discover, and American Express also accepted. Please include your card number, its expiration date, and your signature on your request. Please indicate which issue(s) of DOS Resource Guide you are ordering.

TOLL-FREE 1-800-349-7327 TO ORDER!

Send your check or money order to: DOS Resource Guide P.O. Box 672 Mount Morris, IL 61054

The Debug script DATECHK.SCR creates DATECHK.COM, a program that can determine quickly whether a monthly computing chore should be performed.

```

N C:\DATECHK.COM
A 100
PUSH CS
POP DS
MOV AH,4E
MOV DX,0140
MOV CX,0000
INT 21
JB 012D
MOV DX,[0098]
MOV CL,05
ROR DX,CL
AND DX,000F
MOV [0150],DL
MOV AH,2A
INT 21
CMP DH,[0150]
JNZ 012D
MOV AX,4C00

INT 21
MOV DX,0140
MOV CX,0000
MOV AH,3C
INT 21
MOV AX,4C01
INT 21

F 13C 13F 90
A 140
DB "C:\DATE.FIL"

F 14B 152 00
RCX
52

W
Q
End

```

lated and that file is an 82-byte COM file. If you employ a sufficiently large disk cache and use a disk optimizer to group all your start-up files (including DATECHK.COM), my program will save you time at bootup.

—Marc Scheffler

SWAPPING PRINTER PORTS

Most DOS programs and the Print Screen function assume that you want to print to your LPT1 port, but

Use this Debug script, LPTSWAP.SCR, to create LPTSWAP.COM, a utility that lets you swap parallel printer ports from the DOS command line or from within a batch program.

```

N LPTSWAP.COM
A 100
POP DS
MOV SI,408
LODSW
XCHG AX,[SI]
MOV [408],AX
INT 20

RCX
C
W
Q

```

End

that may not always be the case. If LPT1 is hooked to a network printer, you may have a second printer on LPT2. Or you may have a laser printer on LPT1 but keep your old dot-matrix on LPT2 for printing labels.

DOS doesn't provide a way to switch printer port designations. And you're certainly not going to crawl behind your PC and switch cables every time you want to switch printers! The solution is a simple utility that switches the addresses of the printers internally.

LPTSWAP.SCR is a Debug script file that creates the utility LPTSWAP.COM. Use Edit or Edlin to type in and save LPTSWAP.SCR, being sure to include the blank line before the RCX instruction.

Now, use the following command to create LPTSWAP.COM:

```
DEBUG < LPTSWAP.SCR
```

DOS then returns you to the DOS prompt.

From now on, whenever you want to swap printer ports, simply type LPTSWAP. To restore the original assignments, run LPTSWAP.COM again.

—Robert L. Hummel

A PC TYPEWRITER

Although the PC marketplace offers plenty of sophisticated software, you may find times when it isn't worth the trouble to load your word processor simply to address an envelope or type a quick note. If pencil and paper just won't do, perhaps you can benefit from the following short batch file, which lets you use your printer as a rudimentary typewriter:

```

@ECHO OFF
CLS
ECHO Press Enter to end each line.
ECHO Press F6 and Enter to quit.
COPY CON PRN
ECHO ^L >PRN
CLS

```

Save the file as TYPWR.BAT, or something similar, to avoid confusion with DOS's TYPE command. To create the ^L character in the sixth line, hold down the Alt key while typing the sequence 012 on the numeric keypad, and release the Alt key.

The COPY CON command in the fifth line lets you type several lines of text from the keyboard. When you press F6 followed by Enter, DOS sends the text to your printer. The sequence ^L in the next line is required for page printers (such as laser printers) that need a form-feed

SETKEY.BAT redefines your computer's function keys. You may use it to run DOS commands or programs by typing a single key or key combination.

```
@ECHO OFF
REM SETKEY.BAT assigns a key or
REM key combination to execute a
REM string of characters. The
REM first parameter identifies the
REM key to which you want to
REM assign a function. The second
REM parameter identifies the
REM function. SETKEY uses the
REM variable KEY to keep a record
REM of all function-key
REM assignments made since the
REM last bootup. KEY is reset each
REM time you boot up or each time
REM you run SETKEY. Because
REM SETKEY.BAT relies on the
REM environment variable KEY to
REM update its text file record,
REM running SETKEY.BAT the first
REM time after a reboot makes the
REM text file inaccurate. You
REM should write down your
REM function key changes.
```

```
IF "%1"==" " GOTO HELP
IF "%1"=="?" GOTO HELP
IF "%1"=="H" GOTO HELP
IF "%1"=="h" GOTO HELP
```

```
REM Re-create SETKEY.TXT, if
REM SETKEY.BAT has not been run
REM this session.
IF NOT "%KEY%"==" " GOTO KEYS
ECHO *** PLEASE NOTE ***
ECHO >C:\SETKEY.TXT
ECHO KEY ASSIGNMENTS MADE THIS
ECHO SESSION WITH SETKEY.BAT
ECHO ARE : >> C:\SETKEY.TXT
ECHO. >> C:\SETKEY.TXT
```

```
GOTO KEYS
```

```
:HELP
PROMPT $E[1;44;33m
ECHO ON
ECHO OFF
CLS
ECHO SETKEY.BAT was written by
ECHO JIM BELL, 11-10-92
ECHO.
ECHO DEVICE=ANSI.SYS must be in
```

```
ECHO your CONFIG.SYS file for
ECHO this program to work. The
ECHO first parameter for the
ECHO ECHO function key must be in
ECHO uppercase. No spaces are
ECHO allowed except the spaces
ECHO preceding each parameter.
ECHO Using a third parameter
ECHO of -M suppresses the display
ECHO of SETKEY.TXT.
```

```
ECHO.
ECHO *** Examples of use ***
ECHO SETKEY F1 DIR/P -M Sets
ECHO F1 to list the directory
ECHO (no message)
```

```
ECHO.
ECHO SETKEY SF1 MENU.BAT
ECHO Sets Shift-F1 to run MENU.BAT
ECHO.
```

```
ECHO SETKEY CF1 123.BAT Sets
ECHO Ctrl-F1 to run Lotus 1-2-3
ECHO.
```

```
ECHO SETKEY AF1 CD\UTIL Sets
ECHO Alt-F1 to change to \UTIL
ECHO
```

```
ECHO.
ECHO When you continue, the
ECHO screen will clear.
```

```
PAUSE
PROMPT $E[0;40;37m $P$G
ECHO ON
ECHO OFF
CLS
GOTO END
```

```
:KEYS
SET KEY=0
IF %1==F1 SET KEY=59
IF %1==F2 SET KEY=60
IF %1==F3 SET KEY=61
IF %1==F4 SET KEY=62
IF %1==F5 SET KEY=63
IF %1==F6 SET KEY=64
IF %1==F7 SET KEY=65
IF %1==F8 SET KEY=66
IF %1==F9 SET KEY=67
IF %1==F10 SET KEY=68
IF %1==F11 SET KEY=133
IF %1==F12 SET KEY=134
IF %1==SF1 SET KEY=84
IF %1==SF2 SET KEY=85
```

```
IF %1==SF3 SET KEY=86
IF %1==SF4 SET KEY=87
IF %1==SF5 SET KEY=88
IF %1==SF6 SET KEY=89
IF %1==SF7 SET KEY=90
IF %1==SF8 SET KEY=91
IF %1==SF9 SET KEY=92
IF %1==SF10 SET KEY=93
IF %1==SF11 SET KEY=135
IF %1==SF12 SET KEY=136
IF %1==CF1 SET KEY=94
IF %1==CF2 SET KEY=95
IF %1==CF3 SET KEY=96
IF %1==CF4 SET KEY=97
IF %1==CF5 SET KEY=98
IF %1==CF6 SET KEY=99
IF %1==CF7 SET KEY=100
IF %1==CF8 SET KEY=101
IF %1==CF9 SET KEY=102
IF %1==CF10 SET KEY=103
IF %1==CF11 SET KEY=137
IF %1==CF12 SET KEY=138
IF %1==AF1 SET KEY=104
IF %1==AF2 SET KEY=105
IF %1==AF3 SET KEY=106
IF %1==AF4 SET KEY=107
IF %1==AF5 SET KEY=108
IF %1==AF6 SET KEY=109
IF %1==AF7 SET KEY=110
IF %1==AF8 SET KEY=111
IF %1==AF9 SET KEY=112
IF %1==AF10 SET KEY=113
IF %1==AF11 SET KEY=139
IF %1==AF12 SET KEY=140
IF %KEY%==0 GOTO HELP
```

```
:ADDTXT
ECHO %1 has been set to "%2" >>
C:\SETKEY.TXT
PROMPT $E[0;%KEY%;"%2";13p
ECHO ON
ECHO OFF
IF "%3"=="-M" GOTO END
CLS
TYPE C:\SETKEY.TXT
```

```
:END
PROMPT $P$G
```

End

character (ASCII 12) to print and eject a page.

—John Wolfskill

FUNCTION KEYS AND ANSI.SYS

You don't need DOSKEY to reduce the amount of typing you do at the DOS prompt. SETKEY.BAT lets you assign commands to your system's 48 function keys and execute a DOS command or run a batch file at the touch of a single key or a Shift-, Alt-, or Ctrl-key sequence. A single keystroke might, for instance, format a floppy disk, backup the files on your hard disk, or run a batch file that loads your favorite program. On my system, I have assigned a function key to execute the command DIR /ON /P. When I press the correct function key, DOS displays a directory in alphabetical order by filename, one screen page at a time.

To run SETKEY.BAT, you must add a DEVICE=ANSI.SYS line to your CONFIG.SYS file and reboot to have this change take effect. In addition to changing your computer's function-key assignments, SETKEY.BAT writes a text file, called SETKEY.TXT, to the root directory of drive C. This file, which keeps track of any key assignments you make, is displayed each time you assign a new function key. If you call SETKEY.BAT from other batch files, you should suppress the display of this file by adding the parameter -M to the SETKEY command. (-M is case sensitive.)

SETKEY.BAT is versatile in that it lets you reassign keys on the fly or from other batch files. It also maintains a self-clearing record of key assignments. Another nice feature of SETKEY.BAT is that it doesn't override the function-key assignments you establish with DOSKEY.

The batch file includes a help section that appears when you type SETKEY and follow it with an H, h, or ?. SETKEY.BAT also displays its help information if you forget to specify a parameter or use an invalid one.

—James A. Bell

TOGGLING NUM LOCK REMOTELY

From the computer's point of view, your keyboard has four shift-state keys: Num Lock, Scroll Lock, Caps

The Debug script, NLON.SCR, creates the executable program NLON.COM, which turns on the Num Lock key. To modify the program to turn on the other status keys, change the fourth line as described in the accompanying text.

```
N NLON.COM
A 100
POP DS
OR BY [417],20 ;Alter this line
INT 20
```

```
RCX
8
W
Q
```

End

The Debug script, NLOFF.SCR, creates the executable program NLOFF.COM, which turns off the Num Lock key. To modify the program to turn off the other status keys, change the fourth line as described in the accompanying text.

```
N NLOFF.COM
A 100
POP DS
AND BY [417],DF ;Alter this line
INT 20
```

```
RCX
8
W
Q
```

End

Lock, and Insert. The computer's BIOS (Basic Input/Output System) keeps track of these four keys by setting a single bit in memory when each is turned on, and by clearing the bit when each is turned off.

If a program turns one of these status bits on or off, the BIOS notices the change, sets the keyboard status lights appropriately, and does some other housekeeping. In effect, the program and the BIOS, working together, press the status key for you. The status bits are all kept in the same byte in memory, along with other keyboard information.

I've written a short script file, NLON.SCR, which you may use to create NLON.COM, a program that can turn on the Num Lock key for you. You may modify the script file to create programs to turn on the other status keys. A slightly different script file, NLOFF.SCR, lets you create NLOFF.COM, a program that turns off the Num Lock key. As with NLON.SCR, you may modify NLOFF.SCR to create programs that turn off the other status keys. To create NLON.COM and NLOFF.COM, type in the Debug scripts as shown in the listings and assemble them by typing DEBUG < NLON.SCR or DEBUG < NLOFF.SCR at the DOS prompt. (Don't type the semicolon or the comment in the fourth line, and be sure to type the blank line after the INT 20 instruction.)

To create a program that turns on the Scroll Lock key, substitute the value 10 for the value 20 in the fourth line of NLON.SCR and then assemble the program as SLON.COM. (Be sure to change the program's name in the first line of the listing.) To turn on the Caps Lock, use 40; to turn on the Insert key, use 80. To create programs that turn off the Scroll Lock, Caps Lock, and Insert keys, substitute—respectively—EF, BF, and 7F for DF. ■

Order **THE COMPANION DISK** Now!
Or Reserve **EXTRA COPIES** of

MAXIMUM DOS

**HIGH-POWERED
TOOLS & TECHNIQUES**
to tune your PC for **MAXIMUM PERFORMANCE**

HOW TO:

- ★ Fine-tune AUTOEXEC.BAT and CONFIG.SYS to free up memory, load your favorite programs, and customize your PC setup.
- ★ Run DOS's own disk utilities to troubleshoot your hard drive and make it work faster.
- ★ Squeeze the last byte out of memory using MEMMAKER and EMM386.
- ★ Speed up hard drive reads and writes with SMARTDrive, DOS's top-notch caching utility.

THE MAXIMUM DOS TOOL KIT:

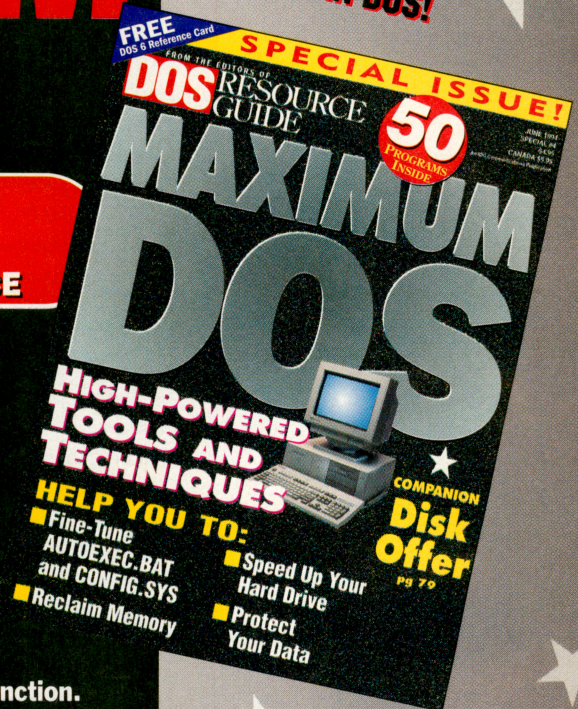
- Dozens of DRG's best utilities reprinted and organized by function.
- Batch files and QBasic programs for taming disks and directories.
- Hardware Helpers: get control of the keyboard, cursor, and screen.
- Save typing time; all listings are on the disk!

DON'T MISS THE COMPANION DISK

It's jam-packed with over 50 of our best batch files, Debug scripts, and QBasic programs, all designed to make you work faster and smarter. The same great programs from the magazine!

BONUS—Our experts' favorite shareware: **4DOS 5.0**, a perennial best-seller that provides power and flexibility at the command line. **HYPERDisk 4.65**, a disk cache that speeds up your hard drive by placing frequently accessed data in RAM. **LIST 9.0e**, the classic replacement for DOS's TYPE command lets you display the contents of files easily.

**EVERYTHING YOU NEED
IS ALREADY
IN DOS!**



**FREE
REFERENCE CARD**

to DOS 6's built-in
memory & disk utilities

YES! I WANT MAXIMUM DOS!
PLEASE SEND ME:

Qty	
	BOTH MAXIMUM DOS and the COMPANION DISK for \$14.90
	Just the MAXIMUM DOS SPECIAL ISSUE for \$4.95
	Just the COMPANION DISK for \$13.95

3.5" 5.25"

Please Circle
Disk Size

3.5" 5.25"

Total Enclosed: ☐

Canada, Mexico, and Foreign Surface add \$5.50
per issue/disk ordered (includes applicable tax).
Airmail add \$9.50. Prepayment is required on all orders.
* U.S. funds on U.S. banks only. Magazines and disks may be shipped separately. Please allow 3-4 weeks for delivery.

AS454

☐ Check or money order enclosed (payable to DOS Resource Guide)
Charge my: ☐ VISA ☐ AmEx ☐ Discover ☐ MC
Card # _____
Signature _____
Name _____
Address _____
City _____ State _____ Zip _____
Exp. Date _____

MAIL TO: DOS Resource Guide,
P.O. Box 672, Mt. Morris, IL 61054
Or Call:

1-800-349-7327

DOS RESOURCE GUIDE

DOS Resource Guide is a publication of International Data Group, the world's largest publisher of computer-related information and the leading global provider of information services on information technology. International Data Group publishes over 194 computer publications in 62 countries. Forty million people read one or more International Data Group publications each month. International Data Group's publications include: ARGENTINA'S *Computerworld Argentina*, *Infoworld Argentina*; ASIA'S *Computerworld Hong Kong*, *PC World Hong Kong*, *Computerworld Southeast Asia*, *PC World Singapore*, *Computerworld Malaysia*, *PC World Malaysia*; AUSTRALIA'S *Computerworld Australia*, *Australian PC World*, *Australian Macworld*, *Network World*, *Reseller*, *IDG Sources*; AUSTRIA'S *Computerwelt Oesterreich*, *PC Test*; BRAZIL'S *Computerworld*, *Mundo IBM*, *Mundo Unix*, *PC World*, *Publish*; BULGARIA'S *Computerworld Bulgaria*, *Ediworld*, *PC & Mac World Bulgaria*; CANADA'S *Direct Access*, *Graduate Computerworld*, *InfoCanada*, *Network World Canada*; CHILE'S *Computerworld*, *Informatica*; COLOMBIA'S *Computerworld Columbia*; CZECH REPUBLIC'S *Computerworld*, *Elektronika*, *PC World*; DENMARK'S *CAD/CAM WORLD*, *Communications World*, *Computerworld Danmark*, *LOTUS World*, *Macintosh Produktkatalog*, *Macworld Danmark*, *PC World Danmark*, *PC World Produktguide*, *Windows World*; ECUADOR'S *PC World*; EGYPT'S *Computerworld (CW) Middle East*, *PC World Middle East*; FINLAND'S *MikroPC*, *Tietoviikko*, *Tietoverkko*; FRANCE'S *Distributique*, *GOLDEN MAC*, *InfoPC*, *Languages & Systems*, *Le Guide du Monde Informatique*, *Le Monde Informatique*, *Telecoms & Reseaux*; GERMANY'S *Computerwoche*, *Computerwoche Focus*, *Computerwoche Extra*, *Computerwoche Karriere*, *Information Management*, *Macwelt*, *Netzwelt*, *PC Welt*, *PC Woche*, *Publish*, *Unit*; HUNGARY'S *Alaplap*, *Computerworld SZT*, *PC World*; INDIA'S *Computers & Communications*; ISRAEL'S *Computerworld Israel*, *PC World Israel*; ITALY'S *Computerworld Italia*, *Lotus Magazine*, *Macworld Italia*, *Networking Italia*, *PC World Italia*; JAPAN'S *Computerworld Japan*, *Macworld Japan*, *SunWorld Japan*, *Windows World*; KENYA'S *East African Computer News*; KOREA'S *Computerworld Korea*, *Macworld Korea*, *PC World Korea*; MEXICO'S *Compu Edicion*, *CompuManufactura*, *Computacion / Punto de Venta*, *Computerworld Mexico*, *MacWorld*, *Mundo Unix*, *PC World*, *Windows*; THE NETHERLAND'S *Computer! Totaal*, *LAN Magazine*, *MacWorld*; NEW ZEALAND'S *Computer Listings*, *Computerworld New Zealand*, *New Zealand PC World*; NIGERIA'S *PC World Africa*; NORWAY'S *Computerworld Norge*, *C / World*, *Lotusworld Norge*, *Macworld Norge*, *Network*, *PC World Ekspress*, *PC World Norge*, *PC World's Product Guide*, *Publish World*, *Student Data*, *Unix World*, *Windowsworld*; IDG *Direct Response*; PANAMA'S *PC World*; PERU'S *Computerworld Peru*, *PC World*; PEOPLE'S REPUBLIC OF CHINA'S *China Computerworld*, *PC World China*, *Electronics International*, *China Network World*; IDG HIGH TECH BEIJING'S *New Product World*; IDG SHENZHEN'S *Computer News Digest*; PHILLIPPINE'S *Computerworld*, *PC World*; POLAND'S *Computerworld Poland*, *PC World / Komputer*; PORTUGAL'S *Cerebro / PC World*, *Correio Informatico / Computerworld*, *MacIn*; ROMANIA'S *PC World*; RUSSIA'S *Computerworld-Moscow*, *Mir-PC*, *Sety*; SLOVENIA'S *Monitor Magazine*; SOUTH AFRICA'S *Computing S.A.*; SPAIN'S *Amiga World*, *Computerworld Espana*, *Comunicaciones World*, *Macworld Espana*, *NeXTWORLD*, *PC World Espana*, *Publish*, *Sunworld*; SWEDEN'S *Attack*, *ComputerSweden*, *Corporate Computing*, *Lokala Natverk / LAN*, *Lotus World*, *MAC&PC*, *Macworld*, *Mikrodatorn*, *PC World*, *Publishing & Design (CAP)*, *Datalingenjoren*, *Maxi Data*, *Windows World*; SWITZERLAND'S *Computerworld Schweiz*, *Macworld Schweiz*, *PC & Workstation*; TAIWAN'S *Computerworld Taiwan*, *Global Computer Express*, *PC World Taiwan*; THAILAND'S *Thai Computerworld*; TURKEY'S *Computerworld Monitor*, *Macworld Turkiye*, *PC World Turkiye*; UKRAINE'S *Computerworld*; UNITED KINGDOM'S *Lotus Magazine*, *Macworld*, *Sunworld*; UNITED STATES' *AmigaWorld*, *Cable in the Classroom*, *CD Review*, *CIO*, *Computerworld*, *Desktop Video World*, *DOS Resource Guide*, *Electronic News*, *Federal Computer Week*, *Federal Integrator*, *GamePro*, *IDG Books*, *Infoworld*, *Infoworld Direct*, *Laser Event*, *Macworld*, *Multimedia World*, *Network World*, *NeXTWORLD*, *PC Games*, *PC Letter*, *PC World*, *Publish*, *Sumeria*, *SunWorld*, *SWATPro*, *Video Event*; VENEZUELA'S *Computerworld Venezuela*, *MicroComputer-world Venezuela*; VIETNAM'S *PC World Vietnam*.

HOW TO USE THIS CARD

OUR READER SERVICE CARD HELPS YOU acquire more information about software and hardware products advertised in *DOS Resource Guide*.

Subject to the request of the advertiser, each advertisement that appears in *DOS Resource Guide* has a Reader Service Number (RS#) associated with it. To receive more information about a product:

1. Note the number that appears in the advertisement.
2. Circle the corresponding number on our postpaid card, opposite.
3. Be sure to PRINT your name & address on the card, and mail the card today.

DOS Resource Guide

Steve Robbins, Publisher (800) 441-4403

Bob Byrne, National Accounts Mgr. (908) 972-8834

Linda Guyette, Advertising Sales/Production
(603) 924-0130 or (800) 441-4403

Advertiser	RS#	Page
AllMicro, Inc.	167	Cov 3
American Eagle Publications	140	87
Business Computer Systems, Inc.	124	86
EFD Systems.	144	86
FPLot Corporation	143	86
Hi-Tech Components Distributors, Inc.	175	84
Hyperkinetix, Inc.	165	85
Insight Development Corp.	147	86
National Computer Inc.	170	82
NeoSoft Corp.	*	Cov 4
NeoSoft Corp.	*	83
PowerBASIC, Inc.	111	5
Queblo	136	87
SET, Inc.	131	87
TeraTech, Inc.	114	87
WordPerfect Corporation	*	Cov 2, 1

*This advertiser prefers to be contacted directly.

FREE PRODUCT INFO!

**Want to Know
More about
Products or
Services
Advertised
In This Issue?**

Here's How:

Print your name and
address where indicated.

Tell us about yourself by
answering the questions.

Circle the numbers on the
card corresponding to the
reader service number
for products and services
that interest you.

Tear out and mail the card.

**SUBSCRIBE
TO**

DOS Resource Guide!

Simply check off the box
on the reader service card.
We'll bill you \$23.70* for
one year (6 issues).

NO OBLIGATION.

Literature on products and services
will be sent to you directly
from advertisers,

FREE OF CHARGE.

*This rate is limited to the U.S. and its possessions.

DOS RESOURCE GUIDE

June 1994
Expires July 31, 1994

READER SERVICE CARD

NAME/TITLE _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____
TELEPHONE _____
FAX NUMBER _____

100	113	126	139	152	165
101	114	127	140	153	166
102	115	128	141	154	167
103	116	129	142	155	168
104	117	130	143	156	169
105	118	131	144	157	170
106	119	132	145	158	171
107	120	133	146	159	172
108	121	134	147	160	173
109	122	135	148	161	174
110	123	136	149	162	175
111	124	137	150	163	176
112	125	138	151	164	177

BSP4R1

☐ Check here to start your subscription to DOS Resource Guide. You will be billed \$23.70* for a one-year subscription (6 issues).
Is this your copy of DOS Resource Guide? **If YES, Circle #100 If NO, Circle #101**

DOS RESOURCE GUIDE

June 1994
Expires July 31, 1994

READER SERVICE CARD

NAME/TITLE _____
COMPANY _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____
TELEPHONE _____
FAX NUMBER _____

100	113	126	139	152	165
101	114	127	140	153	166
102	115	128	141	154	167
103	116	129	142	155	168
104	117	130	143	156	169
105	118	131	144	157	170
106	119	132	145	158	171
107	120	133	146	159	172
108	121	134	147	160	173
109	122	135	148	161	174
110	123	136	149	162	175
111	124	137	150	163	176
112	125	138	151	164	177

BSP4R2

☐ Check here to start your subscription to DOS Resource Guide. You will be billed \$23.70* for a one-year subscription (6 issues).
Is this your copy of DOS Resource Guide? **If YES, Circle #100 If NO, Circle #101**

YOU Be the Editor!

The editors at **DOS Resource Guide** are always looking for ways to improve coverage. The following is a list of topics we do not currently cover, but are considering adding to the magazine. Please indicate your interest in each topic on a 1 to 4 scale by circling the appropriate number.

	Great Interest		No Interest	
Application tips	4	3	2	1
Game playing tips	4	3	2	1
Hardware reviews	4	3	2	1
Novell DOS 7 column	4	3	2	1
Software reviews	4	3	2	1
Upgrade news	4	3	2	1
Utility tips	4	3	2	1
Windows column	4	3	2	1
Windows tips	4	3	2	1
Other (please list) _____				

☐ Check here if you are a subscriber

SP4-94

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 1007 PITTSFIELD, MA

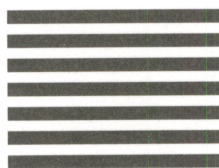
POSTAGE WILL BE PAID BY ADDRESSEE

DOS Resource Guide

Attn: Reader Service Department

PO Box 5384

Pittsfield MA 01203-9733

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATES
BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 1007 PITTSFIELD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

DOS Resource Guide

Attn: Reader Service Department

PO Box 5384

Pittsfield MA 01203-9733

NO POSTAGE
NECESSARY
IF MAILED
IN THE UNITED
STATESPLACE
STAMP
HERE
DOS Resource Guide

80 Elm Street

Peterborough NH 03458

800-349-7327

Subscribe Today! 800-349-7327 Save up to 25% Off the Cover Price


```

LIST 1 89% 02/04/94 09:56 * C:\AUTOEXEC.BAT
C:\NDW\IMAGE.EXE
PATH C:\WINDOWS;C:\NDW;C:\DOS;C:\WINWORD;C:\;C:\BATCH;C:\NORTON;C:\UTILS;C:\ZIP2
nav c:\
USAGE
C:\DOS\SMARTDRV.EXE
verify on
set prompt=$P -$G
set winpmt=[s+[f+[0:30:46m+[K DOS Session Under Windows+[15CAIt+Tab to Switch:
set USERID=JDD
set tmp=c:\temp
set temp=c:\temp
LIST 1 88% 03/01/94 11:51 * C:\MAXDOS\MANUAL.DOC
*****
* DOS Resource Guide *
* Maximum DOS Companion Disk *
* Instruction Manual *
*****

What's on this Disk?

Your Maximum DOS Companion Disk includes program listings for all of the batch
files, Debug scripts, and QBasic programs presented in Maximum DOS, the
Command> *** Top-of-file *** Keys: F1=< PgUp PgDn F10=exit F1=Help

```

Photo. LIST PLUS is a shareware program that allows you to view the contents of files by scrolling through a directory. You can also use it to view more than one file at a time, as shown above.

(Continued from page 88)

mand, to point-and-shoot file selection for any command, to the ability to completely redefine your system's commands. Yet 4DOS is fully compatible with COMMAND.COM. It provides enhancements to most of the DOS commands, as well as more than 40 new commands. These improvements make 4DOS a much richer and more powerful working environment than COMMAND.COM, without sacrificing the compatibility, flexibility, and control you get from working at the DOS prompt.

4DOS speeds up your existing DOS batch files and introduces a new kind of batch-file processing that is 5 to 10 times faster than DOS alone. If you're tired of the limitations of COMMAND.COM, you'll love 4DOS!

HyperDisk SpeedKit 4.65

HyperDisk SpeedKit comprises three PC speed-up utilities that dramatically increase the efficiency of your computer. You can use them together or separately to customize your PC to fit your personal work style.

HyperDisk is an extremely efficient disk-caching utility. It can replace the SMARTDrive program that comes with DOS. HyperDisk can improve hard disk performance by up to 1,000 percent. At the same time, it prolongs the life of your drive by reducing the wear and tear caused by frequent and redundant disk accesses.

HyperKey is a powerful keyboard enhancement package that lets you customize all keyboard functions, including the type-ahead buffer size, key repeat rate, length of delay before a key begins repeating, keyboard clicking sounds; it also includes Touch Shifting for one-finger typing.

HyperScreen is a video enhancement tool that helps

you get the most from your existing video hardware. HyperScreen provides automatic screen blanking to keep images from burning into your display, manual screen blanking at the touch of a key to maintain the confidentiality of on-screen documents, and video Basic Input/Output System (BIOS) speedup to increase display speed in all text modes.

LIST PLUS 9.0e

LIST is a file viewing and browsing utility. You use LIST to display files line by line with the aid of scrolling, positioning, and filtering commands.

LIST PLUS—the version on the Companion Disk—has additional file management commands that go beyond the usual file viewing and browsing, including file copying, moving, renaming, and deleting. LIST PLUS allows multiple windows (see Photo) so you can view more than one file at a time. You also can start another program from within LIST, then return to LIST when the program exits.

Also on the disk are LISTS and LISTR, the “small” and “regular” versions of LIST that provide fewer features, but require much less memory. LISTS runs in as little as 30K of memory, while LISTR needs only 80K. ■

SHAREWARE INFORMATION

4DOS 5.0, \$69

(\$6 shipping and handling)
JP Software Inc.
P.O. Box 1470
East Arlington, MA 02174
617-646-3975

HyperDisk SpeedKit 4.65, \$49.95

(\$2.95 shipping and handling)
HyperWare
185 Berry Street
San Francisco, CA 94107
415-882-1740

LIST Plus 9.0e, \$37

(\$3 shipping and handling)
Vernon D. Buerg
139 White Oak Circle
Petaluma, CA 94952
707-778-1811

Reality Check: value is defined by quality. We Guarantee faultless reality.

Reality. That's the period of time that follows the purchase. It can be many things...a nightmare of the unexpected or a time of great satisfaction. And, it all hinges on quality components and the people who back them.

Tap our experience at National Computer and you're assured of receiving 100% pre-tested hardware, hassle-free service and direct tech support. All at surprisingly competitive pricing. Call us first...get your own "reality check" before you buy.

3-YEAR WARRANTY PLUS

Within 48-hours National Computer will repair, 100% FREE of charge*, any component for up to 3-years after purchase. PLUS! Full coverage continues on labor up to 5-years from purchase date.

* Shipping costs will be absorbed by National Computer on any service returns for 1-year.

1-800-896-1133

9a.m. to 9p.m. - 7 days a week.



All 486 systems come with Intel® processors.

486 DX 33 MHz

- Intel® 486 DX 33 MHz Processor
- OPTI Chipset, 2 Vesa Local Bus Slots
- 256k Copy Back Direct Mapped Cache
- 6, 16 Bit and 2, 8 Bit Expansion Slots
- 4 MB Ram (Expandable to 32 MB)
- 1.2 and 1.4 Teac® Floppy Drives
- 212 MB Caviar® Hard Drive
- 1 MB High Speed Video Card
- 2 Serial, 1 Parallel and 1 Game Port
- 14" .28 SVGA NI Touch Monitor
- 101 Enhanced Keyboard
- 3-Button Mouse
- Mini Tower or Desktop Case

only \$1499.00

Includes Microsoft® DOS® 6.0 and Windows™ 3.1 with complete manuals.

- As above with Intel® 486 DX 50 \$1629.00
- As above with Intel® 486 DX 2/50 \$1549.00
- As above with Intel® 486 DX 2/66 \$1689.00

486 SX 25 MHz

- Intel® 486 SX 25 MHz Processor
- OPTI Chipset
- 128k Copy Back Direct Mapped Cache
- 6, 16 Bit and 2, 8 Bit Expansion Slots
- 4 MB Ram (Expandable to 32 MB)
- 1.2 and 1.4 Floppy Drives
- 212 MB Caviar® Hard Drive
- 1 MB High Speed Video Card
- 2 Serial, 1 Parallel and 1 Game Port
- 14" .28 SVGA NI Touch Monitor
- 101 Enhanced Keyboard
- 3-Button Mouse
- Mini Tower or Desk Top Case

only \$1299.00

Includes Microsoft® DOS® 6.0 and Windows™ 3.1 with complete manuals.

- As above with Intel® SX 33 MHz \$1369.00

386 SX 33 MHz

- 386 SX 33
- 2 MB 70 NS Ram
- 256k Trident® Video Card
- 1.44 MB Floppy Disk Drive
- IDE Hard Drive 105 MB NEC
- 2 Serial, 1 Printer and 1 Game Port
- 14" Touch VGA Monitor .39
- 101 Key Enhanced Keyboard
- 3-Button Mouse
- Mini Tower or Desk Top Case

only \$739.00

386 DX 40 MHz

- 386 DX 40 with 128k Cache, OPTI Chipset, 8 Slots
- 2 MB 70 NS Ram
- 512k Trident® Video Card
- 1.44 MB Floppy Disk Drive
- IDE Hard Drive 105 MB NEC
- 2 Serial, 1 Printer and 1 Game Port
- 14" Touch VGA Monitor .39
- 101 Key Enhanced Keyboard
- 3-Button Mouse
- Mini Tower or Desk Top Case

only \$859.00

- Call for prices on Pentium by Intel®
- We'll custom-build your system. Call.
- Call for configurations using 486 Cyrix SX 33"

MOTHERBOARDS

W/CPU	
386 SX 33	80.00
386 SX 40 Sank OK	85.00
386 DX 40 AMI OPTI 128K	125.00
486 SX 25 W/Intel® CPU 128K AMI OPTI	175.00
486 SX 33 W/Intel® CPU 128K AMI OPTI	245.00
486/33 MHz/128 KB with Cyrix 486/33/SX CPU	175.00
486/33 MHz/256 KB with Cyrix 486/33/SX CPU	199.00

W/O CPU	
486 DX AMI VIA 256K ISA	105.00
3/486 AMI OPTI 256K 2 VLB slots	125.00
486 DX AMI 33/50/66 MHz 256K W/Zif	135.00
486 DX AMI 50 MHz 256K Eisa Vesa Zif	270.00

CPUs

Intel® 486 DX 33	275.00
Intel® 486 DX 50	425.00
Intel® DX 2/50	355.00
Intel® DX 2/66	475.00
Cyrix 486/SX 33 CPU pin compatible with Intel®	CALL
486 clip on cooling fans	15.00
Call for math COs and current ram prices	

VIDEO CARDS

Trident® 256K	38.00
Trident® 9000 512K	45.00
Trident® 8900 CL 1024K	55.00
Diamond Stealth® 24X 1 MB ISA 32K color	156.00
Orchid Fahrenheit® 1280 1024 ISA	153.00
Diamond Stealth® Pro 1 MB VLB Ace	255.00
Diamond Stealth® Pro 2 MB VLB Ace	355.00
Diamond Viper® 2 MB VRAM VLB Ace	395.00

HDDs

212 MB Western Digital Caviar® 14 MS	210.00
250 MB Western Digital Caviar® 13 MS	226.00
340 MB Western Digital Caviar® 13 MS	285.00
130 MB Seagate® ST3144A	177.00

We carry the full line of Fujitsu HDDs, call for current pricing.

Call for specials on all Seagate® & Western Digital® drives.

FDDs

Epson® or Teac® 1.44 3.5"	44.00
Epson® or Teac® 1.2 5.25"	47.50
Mitsumi® 1.44 3.5"	40.00
Mitsumi® 1.2 5.25"	44.00
Teac® dual drive	125.00

CD-ROM DRIVES

Mitsumi® 350 MS w/interface card	175.00
Sony 31A w/interface card	185.00
NEC 84 280 MS INT	499.00
Toshiba® XM3401-S 200 MS photo CD comp	495.00
New Mitsumi® CRMC-FX001D double speed	CALL

MULTI MEDIA

Sound Blaster® Deluxe 8 Bit w/software	99.00
Sound Blaster® Pro Deluxe w/software	130.00
Sound Blaster® 16 Bit True Stereo w/software	175.00
Sound Blaster® 16 Bit w/Asp hardware & software	245.00
Discovery® 8 Bit internal CD ROM w/software	425.00
Edutainment® 8 Bit internal CD ROM w/software	525.00
Discovery® 16 Bit internal CD ROM w/software	425.00
Edutainment® 16 Bit internal CD ROM w/software	565.00
Video Blaster® multimedia interface 16 Bit slot for Windows™	385.00

Video Spigot® digital video capture solution for Windows™ 457.00
Call for updated prices and other sound products.
We carry joysticks from Thrustmaster® and Gravis®!!!
Call for info!!!

MODEMS

Intel® PCFM7500 int. 144/144-14,400bps	
V.42bis modem, 14,400 fax	190.00
Intel® PCFM same as PCFM7500 but external	220.00
Internal 14,400 fax modem w/Winfaxile GVC	165.00
External 14,400 fax modem w/Winfaxile GVC	175.00
US Robotics® Sportster int. 9600bps	
FAX-V.32, V.42 MNP5	189.00
US Robotics® Sportster ext. 9600bps	
FAX-V.32, V.42 MNP5	199.00

I/O CONTROLLERS

Super IDE 2 FDD 2 HDD 2 serial 1 parallel 1 game	15.50
VLB caching controller (up to 16 MB)	205.00
VLB Super IDE 2 FDD 2 HDD 2 serial	
1 parallel 1 game	40.00
ZyXEL® I/O 2 serial 1 parallel 1 game 16450 UART	42.50

TAPE BACKUPS

120 MB Colorado® internal	152.00
250 MB Colorado® internal	182.00
120 MB Colorado® tracker	288.00
250 MB Colorado® tracker	335.00
250 MB Conner® internal	169.00
250 MB Colorado® OEM internal	160.00

CASES

We have many high quality cases. Call for specifications!!!

45 DAY SATISFACTION OR MONEY BACK
LIFETIME 1-800 TECHNICAL SUPPORT
SHIP VIA: FEDERAL EXPRESS, UPS,
AIRBORNE EXPRESS



ZyXEL
The Intelligent Modem



**NATIONAL
COMPUTER**
INCORPORATED

1400 SIXTH STREET S.W., MASSILLON, OHIO 44647
(216) 830-2620 • FAX (216) 830-2615

- All trademarks are the property of their respective owners
- Not responsible for typographical errors
- Prices are higher in our retail store
- Prices subject to change without notice

NeoSoft's Graphical DOS Applications:

When It's Worth Doing Well

9:00



Arrive at work. I turn on my computer to be greeted by the soothing graphical display of my customized **QuikMenu** screen (only \$35*). I notice that I have an E-Mail message. Its from the boss—he wants a catalog containing our newest product, and he needs it tomorrow!

To make a good impression, I decide on an electronic catalog. I use **NeoPaint** (just \$45*) to quickly edit the same images we normally use with our desktop publishing program to produce printed flyers. **NeoPaint** lets me get the results I want—without bringing my computer to a standstill or forcing me to think like a programmer.

9:05



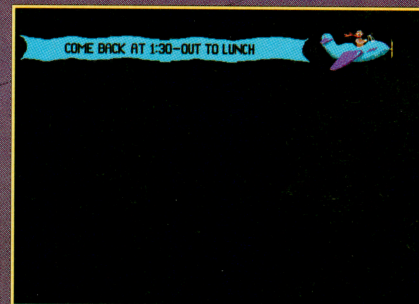
11:15



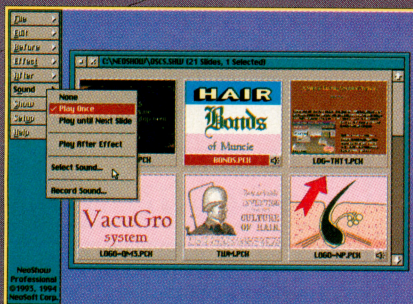
Next, I begin to set up my publication using **NeoBook** (\$45*). I simply drag out frames to contain my illustrations and text files. I can even edit the text using my favorite word processor from within **NeoBook**! Adding command buttons, titles, fonts, rules and other graphical elements is quick and easy.

It's time for lunch, and I leave a message on my **PhantomScreen** screen saver (just \$25*) for DOS. Since I use **PhantomScreen's** password protection feature, I can be confident that no one else is snooping through my system while I'm gone.

12:00



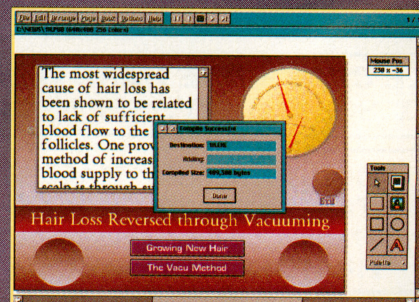
1:30



I decide to create a small animated presentation to be run from within my catalog. **NeoShow Pro** (\$89.95*) allows me to link together images and apply special effects between pictures. I can also record narration on my Sound Blaster™ compatible card.

I create a button to run my presentation within **NeoBook** (the Professional version is only \$89.95*), and quickly check to see that everything works as I intend. Finally, I use the compiler to turn my catalog into a stand-alone EXE program, ready for my boss.

4:00



4:15 And I didn't have to resort to Windows or a Mac!

Download any of the above NeoSoft programs from our 24 hour BBS service for a Free 30 day evaluation.

(503) 383-7195 BBS

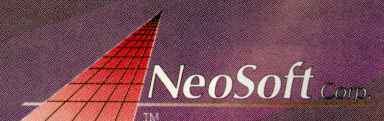
1-800-545-1392 (orders)

(503) 389-5489 TEL.

(503) 388-8221 FAX

* On direct orders, add for shipping/handling: \$5.00 (U.S. Mail) or \$7.00 (FedEx 2nd day) For delivery outside the continental U.S.A.—\$17.50 (Air Mail) or \$21.00 (Express).

All brand and product names are trademarks of their respective holders.



354 NE Greenwood, Suite 108
Bend, OR 97701-4631
U.S.A.

Best Buy Specialist for Over 18 Years

WHOLESALE TO YOU! HARD DRIVES...

60 MB TAPE BACK-UP

3.5" 1/2 HT.

\$59.00 DRIVE ONLY...

\$39.00 with purchase of Hard Drive

MANUFACTURER

MODEL

CAPACITY

FORM FACTOR

SPEED

INTERFACE

PRICE EACH

TWO FOR

THREE FOR

FOUR FOR

*** SEAGATE** ST251 42MB 5.25" 1/2 HT. ... 40MS MFM \$89. \$159. \$224. \$275.

* Recertified

— Quantity Discounts —

IBM 665-053 42MB 5.25" FULL HT. ... 37MS MFM \$69. \$125. \$169. \$215.

Monthly Special!

672MB 16MS \$379.

5.25" FULL HT. / SCSI

FUJITSU M2263 HA

613MB 16MS

\$429.

5.25" FULL HT. / SCSI

SEAGATE ST4702ND

• Archive Internal 150MS TAPE BACK-UP

with 8BIT SCSI CONTROLLER

SYTOS SOFTWARE & DEVICE DRIVERS

\$199. ea.

QUANTITY DISCOUNTS

1.3GB 11.5MS

\$899.

5.25" FULL HT. / SCSI

SEAGATE ST41600N

1.3GB 14.5MS

\$799.

5.25" FULL HT. / ESDI

MICROPOLIS 1518-15

183MB 18MS

\$169.

5.25" HALF HT. / SCSI

SEAGATE ST2209N

525MB 12MS

\$425.

3.5" HALF HT. / SCSI

SEAGATE ST3600N

98MB 16MS

\$119.

3.5" HALF HT. / ESDI

SEAGATE ST1111E

380MB 16MS

\$299.

5.25" HALF HT. / ESDI

MAXTOR XT8380E

80MB 19MS

\$89.

2.5" 1/3 HT. / SCSI

QUANTUM 6080S

100MB 20MS

\$99.

3.5" HALF HT. / SCSI

NEC D3856

122MB 28MS

\$115.

5.25" FULL HT. / RLL

SEAGATE ST4144R

REMOVABLE CARTRIDGE HARD DISK DRIVES:

SYQUEST SQ312 10MB ... 5.25" 1/2 HT. ... \$50. ea.

FLOPPY 360K 5.25" HALF HT. \$13.

DISK 720K 3.5" HALF HT. \$16.

DRIVES: 1.2K 5.25" HALF HT. \$33.

1.44K 3.5" HALF HT. \$30.

— CONTROLLER CARDS —

8 BIT XT MFM HARD ONLY ... \$15.

8 BIT XT RLL HARD ONLY ... \$22.

16 BIT AT RLL HARD ONLY ... \$13.

16 BIT AT MFM HARD ONLY ... \$13.

16 BIT AT MFM HARD / FLOPPY ... \$26.

16 BIT SCSI TMC1660 HARD ONLY ... \$80.

16 BIT SCSI DTC3280 HARD / FLOPPY ... \$90.

16 BIT ESDI OMT18641 HARD ONLY ... \$36.

16 BIT SCSI MCS700 MICROCHANGE ... \$39.

ALL DRIVES NEW / USED OR REFURBISHED TESTED WITH A 90 DAY WARRANTY

HI-TECH

Call Today **(805) 681-9961**

Ask for Drive Sales

COMPONENT DISTRIBUTORS INC.

or FAX your orders: **(805) 681-9971**

Monday through Friday, 7:30 A.M. — 4:30 P.M., PST

59 S. La PATERA LANE • GOLETA, CA 93117 U.S.A.

Circle 175 on Reader Service Card

Quit bitching about batch.

Now you can eliminate 99% of your batch file problems for only **\$99.95**. (Mention this ad to get **33% off** our list price of \$149.95.)

BUILDER™ is the first compiler that transforms sluggish batch files into blazingly fast .EXE files. Not only does it give you turbo language speed, it extends the DOS batch language with almost 200 new commands and keywords.

For example, menu commands like *DropDown*, *PopUp*, and *LightBar* are

built into the language, so you can quickly produce an unlimited number of programs, and distribute as many copies as you wish.

BUILDER also includes a powerful editor to provide an integrated development environment, and code generators to speed your productivity.

And it's the perfect tool for creating installation scripts. **BUILDER** has DOS dexterity that BASIC and C can't match. And its small compiled

code size won't hog memory or disk space.

What's more, by compiling a program with **BUILDER** before you distribute it, you'll have bulletproof security.

Don't keep bitching about batch. Order **BUILDER** today.

Call **1-800-873-9993** for complete details. Visa and Mastercard are welcome. We'll even pay ground shipping. 90 day money back guarantee.

BUILDER

What batch files should have been.

hyperkinetix, inc. 18001 Irvine Blvd., Suite H, Tustin, CA 92680 800/873-9993 714/573-2260 FAX 714/573-2297

Circle 165 on Reader Service Card

DRG MARKETPLACE

READERS: To help you learn more about the many interesting products available for your PC, **DOS Resource Guide** has created the **DRG MARKETPLACE**. This special section makes it easy to contact companies offering a wide range of accessories, software, and services. Listings are grouped by product category for your convenience.

VENDORS: Listings in the **DRG MARKETPLACE** are sold by the column inch. For more information, contact Linda Guyette at **1-800-441-4403** or **603-924-0130**.

SOFTWARE

UTILITIES

**Turn ANY Printer
Into A Plotter
for \$99**

PRINTAPLOT 3.0

- "...it is unbeatable." — CADalyst
Convert HP-GL files
- ▶ Select 255 pens/colors
 - ▶ Change pen width, patterns, colors
 - ▶ Tile "A" to "E" size drawings
 - ▶ Adjust scaling/rotation
 - ▶ Pops up over application
 - ▶ 20 times faster than a pen plotter
 - ▶ Supports over 1,200 printers

800-825-4115

Insight Development
2420 Camino Ramon
San Ramon, CA 94583
Ph: 510-244-2000
Fax: 510-244-2020

Circle 147 on Reader Service Card

PEN PLOTTER EMULATOR

FPlot turns your printer into an HP pen plotter. Fast hi-res, no jagged lines. Vary line width, color. Screen preview--zoom, pan. Works with most CAD programs. Supports most printers. Requires DOS 2.1 or higher. \$119.00 (plus \$3.00 S&H). Visa/MC/Check/MO accepted.

FPlot Corporation

24-16 Steinway Street, Suite 605
Astoria, NY 11103

(718) 545-3505

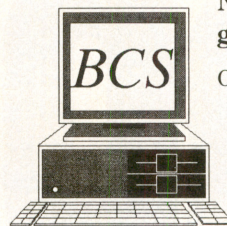
Circle 143 on Reader Service Card

SOFTWARE

BUSINESS OPPORTUNITIES

EARN \$5000 PER MONTH With your home computer!

Spending too much on your computer?
Now it's time to let it earn
good money for you!



Our two successful software packages, **MoneyMaker Volume-1** and **MoneyMaker Volume-2** will give you all the insider information you need to start your own profitable business part-time at home!

Your cost for all this extra income?

*Only \$29.95 for MoneyMaker, Volume-1
Or, order both volumes now, for just \$39.95!!!!*

Includes **FREE** Talking Expert System Demo
and 3 **BIG** catalogs full of more **money-making** ideas.

For 3 catalogs & demo disk, send your check for \$5.
(We always pay postage.)

Please specify disk size. (If unspecified, we ship 3.5")

Send your check, M.O. or American Express to:

BUSINESS COMPUTER SYSTEMS, INC.

46-16 65th Place, Woodside, NY 11377

(718) 898-3126

Circle 124 on Reader Service Card

NEW!
ver. 2.0

DOS LIVES!



Elvis may be dead but DOS ain't! Watch DOS come alive with **IconDOS**; the modern, graphical, icon-based desktop environment for DOS! Uses no resident memory when launching apps from icons so your software won't miss a beat. VGA required.

- * Clock, calendar, calculator
- * Cardfile with phone dialer
- * Notepad, file manager
- * Games, screen saver
- * Icon editor with 500 icons
- * Utility to convert .ICO icons

Shareware ver. available from PC-SIG, PsL, CompuServe & others. For registered ver. send \$24.95 + \$3 S&H, check or m.o. to:



EFD Systems
304 Smokerise Circle
Marietta, GA 30067

05-76627-3026

Circle 144 on Reader Service Card

You've got a **great** product.
But, a small ad budget . . .

Let the
DRG Marketplace
work for you!

MISCELLANEOUS

ON-LINE SERVICE

GET DRG PROGRAMS ON-LINE!

The DOS Resource Guide BBS

Save time and boost your productivity with DOS Resource Guide's electronic bulletin-board system. Every issue's QBasic listings and recommended shareware are available for easy downloading.

Connect time is **FREE**.

You pay only for the phone call. Dial:

1-603-924-3181

(Parity settings are 8-N-1.) After receiving the "Connect" message, press "C" to access the main menu. From there you can choose from among five software categories, or leave us a message. Let us know what's on your mind. And watch for exciting new features on *DOS Resource Guide's* interactive on-line service.

SOFTWARE

POWERBASIC TOOLS

Boost Your BASIC with New ProBas™ 5.6!

New ProBas 5.6 boosts your Basic programs into orbit! And no royalties, ever!

- * Over 900 routines, most in assembly for speed & size! Loads of examples!
- * Arrays, Comms, Strings, Equip, Files, DOS, Memory, Sorts, F/Xs, Virtual windows, GUI,...
- * 1200 pages of docs with detailed tutorial!
- * Fast, flexible code for professional programs!
- * Unlimited free tech support! **Just \$195!**

BasicPro
Readers Choice



Risk FREE 60 day trial!

More QuickBasic, Power Basic, Visual Basic & C tools:

ProMath™ 2.0 - 240 math/engineer/stat/physics/graph fns . .	\$149
FinLib™ 1.1- interest/bond/cashflow/annuity calculations . . .	\$149
ProFont™ 2.1- screen fonts - rotate/color/size/edit/input . . .	\$125
ToolKit™ 3.1- menus, dialogs, B-Tree, EXE patch	\$125
ProVoice™ 1.0- telephone voice processing library	\$199
TeleComm™ 1.2 - modem transfer & terminal emulation	\$75
SpellCheck™ 1.1 - spelling checking library & dictionary . . .	\$49
Ace Comparator™ 1.2- intelligent program diff display	\$99

**FREE demo disk & booklet - call:
800-447-9120 ext. 1087**

TeraTech - Tools for Programmers™

Dept 1087, 100 Park Ave, Suite 360, Rockville MD 20850 USA
Int'l: +1-301-424-3903 Fax: 301-762-8185 BBS: 301-762-8184 9600-N-8-1

Copyright TeraTech 1994. All rights reserved. Trademarks are the property of their respective holders.

Circle 114 on Reader Service Card

MISCELLANEOUS

SUPPLIES

FREE CATALOG

Now you can produce your own customized brochures, presentation materials, newsletters, gift certificates, special announcements and invitations at a fraction of the expense a commercial printer would charge.

Our catalog features



hundreds of laser printer and copier compatible papers, card stocks and supplies designed to create spectacular promotional materials.

All Queblo products are 100% laser printer and copier compatible. Request code #R1028 for your FREE Catalog.

Call Toll Free 1-800-523-9080

QUEBLO 1000 Florida Avenue, Hagerstown, Maryland 21741
24 Hour Fax 1-800-55-HURRY (48779)

Circle 136 on Reader Service Card

MISCELLANEOUS

BOOKS

FREE REPORT

"Computer Viruses and Advanced Operating Systems"

Recently the US virus writing group *Phalcon-Skism* released the first virus for OS/2. Will viruses proliferate like never before as advanced operating systems like OS/2 and Windows NT come on line? Or will they die out? Find out with this free report and get a catalog of unique no-nonsense books and information about viruses. Circle the reader service number, write or call today!

American Eagle Publications, Inc.
PO Box 41401
Tucson, AZ 85717
(800) 719-4957

Circle 140 on Reader Service Card

Did You Know ...

**68% of
DOS Resource Guide
readers use
Windows.**

Say you saw it in the
DRG Marketplace

HARDWARE

VOICE RECOGNITION

THRIFTY TALKER

Real/synthetic speech, music
-- from parallel printer port.
LAP OR DESKTOP, PC/XT ON UP
-- with or without hard disk!

\$19.95 for:

LPT DAC, AM pocket radio
(acts as amplified speaker), and
lots of software, manuals.
Program is compiled BASIC.
Annunciators, language, songs.
For fun or profit.
360K bytes of speech in RAM
-- even on an old PC/XT!

MC/VISA or check (allow for clearance).
Add \$3.50 S&H.

Get two for one S&H charge.
Or, get three, and we pay shipping!

THE THRIFTY TALKER

3205 SE Spyglass Drive
Vancouver, WA 98684
FAX: 206-260-1660

Circle 131 on Reader Service Card

THE MAXIMUM DISK FOR Maximum DOS

Get even more from your *Maximum DOS* issue with our disk containing all the programs in the magazine, plus great shareware for you to try!

by Jeff DeTray

YOU'VE PROBABLY NOTICED THAT THIS SPECIAL *MAXIMUM DOS* ISSUE OF *DOS Resource Guide* contains an unusually large number of program listings. In fact, there are more than 50 batch files, QBasic programs, and Debug scripts in these pages. To save you the time and aggravation of typing and debugging them, we're pleased to offer a special Companion Disk containing every listing in the issue. As an added bonus, we've included for your evaluation three great shareware programs

that epitomize the Maximum DOS concept.

On the Companion Disk, programs from the Tool Kit section of this issue (beginning on page 53) are grouped by function: managing disks and directories, managing files, and hardware helpers. For Debug scripts, we've included both the script files printed in this issue and the executable .COM files they create. Additional programs from other sections of the issue are also on the disk.

By special arrangement with three top vendors, the *Maximum DOS* Companion Disk includes three shareware programs for you to test drive on your own computer. You may install and evaluate these three software packages on your computer for a limited time as specified by the manufacturer of each package, typically 21 to 30 days. At the end of the evalua-

tion period, you either must pay a fee to become a registered user of the program or stop using it altogether. It's your chance to try before you buy.

The cost of the *Maximum DOS* Companion Disk is only \$13.95. Look for the order form on page 79, or call 800-349-7327 for ordering information.

Now here's a brief description of the shareware programs on the disk.

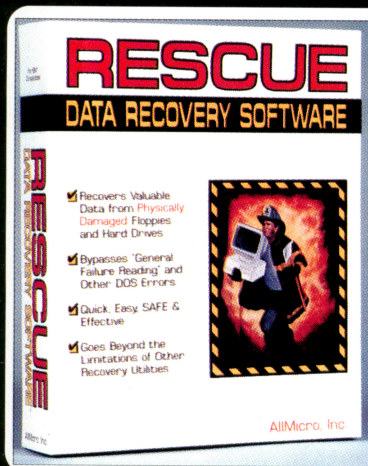
4DOS 5.0

4DOS is designed to help you get the most out of your PC-compatible system. It replaces COMMAND.COM, the command interpreter that comes with all versions of MS-DOS and PC DOS.

4DOS provides a variety of capabilities COMMAND.COM lacks, ranging from a vastly enhanced DIR com-

(Continued on page 81)

FIX ANY PC FAST!



DO-IT-YOURSELF DATA RECOVERY RECOVER DATA FROM PHYSICALLY DAMAGED FLOPPIES & HARD DISKS IN LESS THAN 60 SECONDS!

RESCUE DATA RECOVERY SOFTWARE™ is the **FASTEST, EASIEST & SAFEST** method in data recovery anywhere!

For the first time you can recover a file in less than 60 seconds even when DOS cannot read a physically damaged floppy or hard disk.

Recover Text, .Exe, Graphics files, etc... You can even recover entire sub-directories with a single key stroke.

CALL NOW FOR PRICING!

"In the past, General Failure messages left you few options. Norton, Central Point, or Mace Utilities weren't likely to help... Now there is a better solution... RESCUE can recover data from all but the most thoroughly trashed disk... RESCUE appears to work miracles..."
"Rescue, Version 4.0 offers new features and easy operation that move it out of the guru's toolbox and into the average user's library."
INFOWORLD

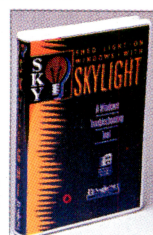


RATED #1

FIX YOUR WINDOWS FAST!

**PC
MAGAZINE
EDITORS'
CHOICE**
July 1993

**BEST
Windows
Trouble-
shooting
Program
Available!**



Skylight™

"... a model of elegance and clarity. Skylight stands out as a product that exudes intelligence in function and design."

- PC Magazine, July 93

TUNE, OPTIMIZE & TROUBLESHOOT WINDOWS for faster speed & top performance!

"Any way you can think of to slice it, Skylight can show it to you-how Windows is using memory, system resources, system metrics, GDI heap usage. This MDI-based program offers an intriguing view into the guts of Windows." - Windows User, June 93

PC DIAGNOSTICS THAT REALLY FIND THE BUGS!

THE TROUBLESHOOTER™ is unlike other diagnostic programs that rely on DOS. The Troubleshooter™ bypasses DOS and tests all major hardware directly for true accuracy while other programs frequently give erroneous test results. Loaded with all the tests you need to quickly and accurately isolate the source of PC failures. Newest version includes low level formatting of I.D.E. drives and cache memory testing. Runs on any PC (with Intel or compatible C.P.U.) independent of any operating system! Priced far below all competitors.

THE DISCOVERY CARD™ is the only tool that truly ends the frustration of IRQ & DMA channel conflicts! Stop wasting time trying to "Figure-it-out" when it comes to setting up add-on cards. Perfect for anyone who upgrades PC's, sets up networks, multimedia, etc...

"If you support PC's, the Discovery Card is likely to become one of your most important tools." - INFOWORLD, September 93

THE POCKET POST™ is the most compact and feature packed diagnostic power-on self test (post) card in the world. Debug dead PC's in less than five minutes! Also, tests power supply and 9 bus/clock signals. Comes with a detachable logic probe.



The Troubleshooter Kit™

CALL NOW FOR PRICING!

NEW!

NO MORE SYSTEM CRASHES

COOL CHIP™ is an inexpensive heat sink that can lower the operating temperature of a C.P.U. by as much as 35°F, thus eliminating system lock-up due to excessive heat.



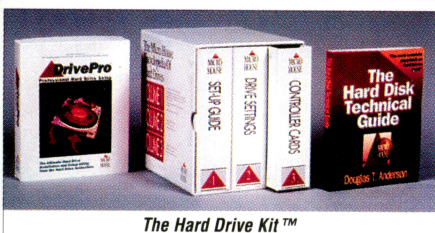
Excessive heat build-up within the C.P.U. is the number one cause of system crashes, lock ups, erratic behavior and intermittent failures.

COOL CHIP™ is the most effective means available to eliminate C.P.U. overheating.

"We had been experiencing various problems with system lock-ups and shut-downs, and we exhausted all other possibilities to resolve these various problems (memory, programs, power supplies, monitors, disk drives)... since the installation of COOL CHIP we have encountered no other problems."

- M.H., Certified Network Engineer

CALL NOW FOR PRICING!



The Hard Drive Kit™

EVERYTHING YOU NEED TO INSTALL, SET-UP & MAINTAIN HARD DRIVES!

DRIVE PRO™ — The all-in-one software utility for the most efficient and correct installation and maintenance of any hard drive! • Install IDE Drives in less than 60 seconds! Automatically sets CMOS, Partitions, and DOS formats without re-booting or

user intervention. • DOS Format any size drive in under 30 seconds! • Drive Table Override allows almost any BIOS to have a user definable drive type. • Drive Boot Fixer is a safe alternative to low level formatting bad IDE Drives. Plus too many other features to mention!

ENCYCLOPEDIA OF HARD DRIVES™ — 3 volumes with over 1500 pages! The largest compiled reference on Hard Drives ever published!

Volume One...SET-UP GUIDE • Interface Types and Installation • Hard Drive Specs for over 2100 drives from 1984 to present.

Volume Two...DRIVE SETTINGS • Explanation of Jumper Types, Changes in Make & Model and Default Jumper Settings • Diagrams for over 1000 drives with detailed specifications.

Volume Three...CONTROLLERS • Over 400 Diagrams of Controllers with detailed specifications. And Much More!

THE HARD DISK TECHNICAL GUIDE™ — Comprehensive field version of the Encyclopedia with over 400 pages of vital specs! Compact to carry in the field.

CALL NOW FOR PRICING!

VITAL HARDWARE SUPPORT AT YOUR FINGERTIPS!

Now available for the first time on CD-Rom, this Technical Library contains complete configurations, specifications, diagrams, settings, component locations, and vital information on 1500 main boards, more than 700 network interface cards, 2100 hard drive models and 400 controller cards. A must for any service department. **CALL NOW FOR PRICING!**



Technical Library on CD-Rom



• Free Tech Support



• Performance Guaranteed



• Next Day Shipping

**ORDER DIRECT — CALL
(800) 653-4933**

AllMicro, Inc.

18820 U.S. Hwy 19 N., #215, Clearwater, FL 34624
INT'L. (813) 539-7283 • Fax (813) 531-0200

Copyright © 1993. Rescue Data Recovery Software™, The Hard Drive Kit™, The Discovery Card™, The Troubleshooter's Kit™, and Cool Chip™ are trademarks of AllMicro, Inc. All Rights Reserved. Other names are trademarks of their associated owners. Specifications subject to change. Some system failures may be beyond repair.

MULTIMEDIA MADE EASY

NeoBook Power Pack

NeoBook's easy to use floating tool bar allows you to construct your publication using simple drag and drop commands.

Easily drop in text files, complete with scroll bars.

It's simple to set up reader command buttons and other interactive controls. Quickly set up an interface which allows the reader to turn pages, jump to another page, pop up messages, run another program, view a slide show presentation, and much more.



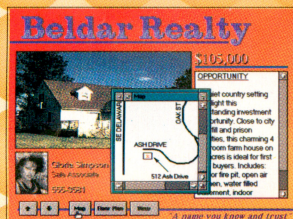
NeoBook compiles your publication into a single executable (EXE) program file for distribution. No run-time royalties.

Import graphic images from the powerful included paint program or from other software.

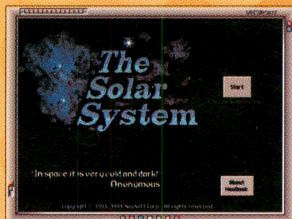
Presentations created with the included slide show program may be played in a window within your publication.

Include sound files. Support for both the PC internal speaker and for Sound Blaster™ compatible audio cards.

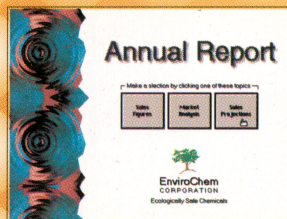
NeoBook Power Pack Can Produce a Variety of Electronic Publications



Promotional Material



Educational Materials



Presentations



Multimedia Newsletters

Ask about our other graphical DOS applications:

NeoPaint Pro Pack

Makes it easy to create and edit graphic images. Includes module which allows images to be arranged into stunning audio-visual presentations. Also available in CD-ROM version, with extensive library of classic art and more.

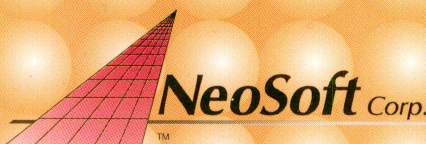
NeoSoft Corp. also carries evaluation copies of several other programs which you may download at no charge from our 24-hour BBS.

QuikMenu III

Puts a beautiful new graphic face on any DOS system. Powerful file manager, desktop utilities, E-Mail, wallpaper, etc. Launches any DOS program with a single mouse click.

For more information,
or a dealer in your area:

1-800-545-1392



All brand and product names are trademarks of their respective holders.

(503) 389-5489 TEL.
(503) 383-7195 BBS
(503) 388-8221 FAX
354 NE Greenwood, Suite 108
Bend, OR 97701-4631
U.S.A.